GEOMETRIC MODELING WITH ALGEBRAIC SURFACES

Chandrajit Bajaj

CSD-TR-825
November 1988

Geometric Modeling with Algebraic Surfaces

Chanderjit L. Bajaj[*]
Department of Computer Science
Purdue University
West Lafayette, IN 47907

November 20, 1988

Abstract

Research in geometric modeling is currently engaged in increasing the geometric coverage to allow modeling operations on arbitrary algebraic surfaces. Operations on models often include Boolean set operations (intersection, union), sweeps and convolutions, convex hull computations, primitive decomposition, surface and volume mesh generation, calculation of surface area and volumetric properties, etc. From these arise a number of basic problems for which effective and robust solutions need to be obtained. We need to devise methods for unambiguous algebraic surface model representations, for converting between alternate internal algebraic curve and surface representations such as the implicit and the parametric, for intersecting algebraic surfaces and topologically analyzing the inherent singularities of their high degree curve components, for sorting points along algebraic curves, for minimum distance and common tangent computations between algebraic curves and surfaces, for containment classifications of algebriac curve segments and algebraic surface patches, etc. Computationally efficient algorithms for all these problems necessitate combining results from algorithmic algebraic geometry, computer algebra, computational geometry and numerical approximation theory. In this paper we present and discuss various such algorithms and approaches for geometric models with algebraic surfaces.

[*]Supported in part by NSF Grant MIP 85-21356, ARO Contract DAAG29-85-C0018 under Cornell MSI and ONR contract N00014-88-K-0402. Invited Paper at "The Mathematics of Surfaces III", Oxford University, Oxford, UK, September 19 - 21, 1988.

1 Introduction

While the geometric capabilities of modeling systems have evolved over the years, current day modelers are nevertheless beset by common problems stemming from the use of inadequate algorithms, ill-conditioned numerical calculations and inconsistent topological decisions. Part of these woes arise from a naive choice of the modeling curve and surface elements. To combat poor approximations and achieve greater accuracy, geometric modeling systems either restrict the algebraic degree of its allowed surfaces to be *planar* [23, 102] at the expense of enormous size complexities of model descriptions [57] ; or *quadrics* [60, 65, 72], however, with the inability of allowing quadric - quadric intersection surface blends [50, 100] ; or parametric patches of various types [25, 32, 35, 46, 47, 63, 68, 78] . The restriction only to parametric surfaces, a subset of algebraic surfaces, leads one to model at times with prohibitively higher degree surface elements than necessary and furthermore the non-closure of parametrics under sweep and convolution operations [19], necessitates approximations for modeling the results of such operations. For example, designers using only parametric surfaces usually jump to the flexibility of bicubic parametric patches (possibly of algebraic degree 18) for various smoothness and tangency requirements, on finding that planar, quadratic and cubic parametric patches (of algebraic degrees 1, 4 and 9, respectivley) are inadequate, see [85]. Not considering general (non-rational) algebraic surfaces of intermediate degree, is clearly too restrictive a design limitation.

In this paper we present the computational viability of geometric operations on models of solid physical objects with arbitrary algebraic surface boundary patches. The class of algebraic surfaces [104], a subset of arbitrary analytic surfaces, provide enough generality to accurately model almost all complex rigid objects. Additionally, algebraic curves and surfaces have compact storage representations and form a class which is complete under all common operations required by a geometric modeling system. Further representing physical objects with algebraic surfaces encompasses all prior modeling approaches [71].

Geometric modeling with algebraic surfaces requires effective and robust algorithms for a wide variety of basic operations. In this paper we present effective methods for unambiguous model representations, for converting between alternate internal representations of algebraic curves and surfaces such as the implicit and the parametric, for computing solutions of systems of polynomial equations, for intersecting algebraic surfaces and analyzing the inherent singularities of their high degree curve components. etc. These algorithms combine techniques from algorithmic algebraic geometry, computer algebra, computational geometry and numerical approximation theory.

Section 2 details definitions and mathematical preliminaries on algebraic curves and surfaces relevant to algorithms in later sections. In Section 3 we present conversion algorithms between implicit and parametric forms as a way of harnessing the advantages of both representations. In Section 4 we consider the computation of solutions of systems of polynomial equations. These arise from the topological tracing of algebraic plane curves, the display of algebraic surfaces, the intersection of two algebraic curves or a pair of algebraic surfaces, the decomposition of algebraic curves into convex segments, the intersection of three algebraic surfaces, etc. In Section 5 we present unambiguous representations of solid models with algebraic surface patch boundaries. These include an algebraic boundary model, a general constructive semi-algebraic description using boolean set operations and a representation of the piecewise tangent space of a convex, algebraic-boundary model. In Section 6 we consider the various algorithms involving numerical calculation and topological reconstruction needed to execute boolean set operations on algebraic boundary models. In Section 7 we deal with various decompostions of a complex, algebraic boundary model into simpler and primitive pieces. In one form it also involves the question of converting from a boundary description to a constructive geometry representation. Section 8 concerns the computation of the smallest enclosing convex model (the convex hull) of an algebraic boundary model. Finally, Section 9 presents algebraic algorithms for computing the sweep and convolution of convex algebraic models and Section 10 highlights the numerous other geometric operations on solid models which are worthy of greater attention and detail.

2 Mathematical Preliminaries

We present some details on the representation of algebraic curves and surfaces, together with some fundamental elimination formulas for systems of polynomial equations. Facts on algebraic curve and surface representations and information on their rationality can be gleaned from numerous books and papers on analytic geometry, algebra and algebraic geometry, see for example [2, 80, 81, 92, 97, 99, 104].

2.1 Algebraic Curves and Surfaces

An algebraic plane curve is implicitly defined by a single polynomial equation $f(x, y) = 0$. A rational algebraic curve can additionally be defined by rational parametric equations which are given as $(x = G_1(u), y = G_2(u))$, where G_1 and G_2 are rational functions of u, i.e., each is a quotient of polynomials in u. Similarly, an algebraic surface is implicitly defined by a single polynomial equation $f(x, y, z) = 0$ and a rational algebraic surface is described by rational parametric equations $(x = G_1(u, v), y = G_2(u, v), z = G_3(u, v))$ where the G_i, $i = 1, 2, 3$, are rational functions. Finally, an algebraic space curve, defined by the intersection of two algebraic surfaces can be given either as a pair of polynomial equations ($f_1(x, y, z) = 0$ and $f_2(x, y, z) = 0$) or as two sets of parametric equations $(x = G_{1,1}(u_1, v_1), y = G_{2,1}(u_1, v_1), z = G_{3,1}(u_1, v_1))$ and $(x = G_{1,2}(u_2, v_2), y = G_{2,2}(u_2, v_2), z = G_{3,2}(u_2, v_2))$, where the $G_{i,j}$, $i = 1, 2, 3$, $j = 1, 2$, are rational functions.

Rational algebraic space curves are additionally representable as $(x = G_1(u), y = G_2(u), z = G_3(u))$, where G_1, G_2 and G_3 are rational functions in u. In modeling the boundary of physical objects it suffices to consider only space curves defined by the intersection of two surfaces. Space curves in general can be defined by several surfaces, however this representation is difficult to handle equationally and one needs then to resort to computationally less efficient ideal-theoretic methods, see [27]. General space curves is a topic with various unresolved issues of mathematical and computational interest and an area of future research, see [1].

Rational curves and surfaces are only a subset of algebraic curves and surfaces of the same degree. We illustrate this with some examples and also figures, see Figure 2.1.1, 2.1.2. Algorithms dealing with rationality is the subject of Section 3. All degree two algebraic curves (conics), are rational. For degree three algebraic curves (cubics): while all singular cubics are rational, the nonsingular cubics only have a parameterization of the type which allows a single square root of a rational function. Only small subsets of degree four and higher algebraic curves are rational. For example, degree four curves (quartics) with a triple point or three double points and degree five curves (quintics) with two triple points or six double points, etc., are rational. In general, a necessary and sufficient condition for the rationality of an algebraic curve of arbitrary degree is given by the Cayley-Riemann criterion: a curve is rational iff $g = 0$, where g, the genus of the curve is a measure of the deficiency of the curve's singularities from its maximum allowable limit [99]. There exist algebraic curves of arbitrary genus, with $\frac{(d-1)(d-2)}{2}$ being the maximum genus for a plane curve of degree d.

It is sometimes useful and efficient to process a given algebraic space curve C by a corresponding plane curve P if there is an invertible map between C and P. Fortunately there exists a birational correspondence between the points of any algebraic space curve C and the points of a corresponding plane curve P, whose genus is the same as that of C, see [1, 91, 99]. Birational correspondence between C and P means that the points of C can be given by rational functions of points of P and vice versa (i.e a 1-1 mapping, except for a finite number of exceptional points, between points of C and P). Hence the genus of a space curve and its criterion for rationality is the same as that of a birationally equivalent plane curve.

Examples of rational algebraic surfaces of degrees two, three and four are illustrated in Figure 2.1.3, 2.1.4. All degree two algebraic surfaces are rational. All degree three surfaces, except the cylinders of nonsingular cubic curves and the cubic cone, have a rational parameterization, with the exceptions again only having a parameterization of the type which allows a single square root of rational functions. Most algebraic surfaces of degree four and higher are not rational, although parameterizable subclasses can be identified. For example, degree four surfaces with a triple point such as the Steiner surfaces or degree four surfaces with a double curve such as the Plucker surfaces are rational. In general, a necessary and sufficient condition for the rationality of an algebraic surface of arbitrary degree is given by Castelnuovo's criterion: $P_a = P_2 = 0$, where P_a is the arithmetic genus and P_2 is the second plurigenus [104].

2.2 Desingularization

Often local information, say near the origin, about an algebraic plane curve $f(x, y) = 0$ curve is useful. The *order form* of f contains information about the curve's behavior at the origin. It is the homogeneous polynomial $F(x, y)$ consisting of the terms of lowest degree in f. When the order form is linear, then the curve f is said to have a *simple* point at the origin. Otherwise the curve is singular at the origin. When the order form is nonlinear then the *order* of the singularity at the origin is the degree of F. Moreover, the linear factors of F are equations of the *tangents* of the curve at the origin. For example, if $f(x, y) = x^4 + 2x^2y^2 + y^4 + 3x^2y - y^3$ then $F(x, y) = 3x^2y - y^3$ and the singularity at the origin is of the third order. Further the equations of the tangents of the curve at the origin are $y = 0$, $\sqrt{3}x - y = 0$ and $\sqrt{3}x + y = 0$. See also Figure 2.2.1 for a graph of the curve f.

Of course non-singular curves are easier to analyze and more tractable in practical applications. Fortunately, there is a method of transforming a curve so that a singular point on it becomes a simple point. Such desingularization of plane curves is based on the following classical Cayley-Riemann Theorem: *Every plane curve can be birationally transformed into a curve devoid of singularities.* For practical applications, there are proofs of this theorem which are constructive and actually derive the needed birational transformation. This process is accomplished by a sequence of elementary quadratic transformations, cf. [3, 99]. The quadratic transformation is:

$$
\begin{aligned}
T:\ x &= r \\
y &= rs
\end{aligned} \tag{1}
$$

with inverse

$$
\begin{aligned}
T^{-1}:\ r &= x \\
s &= y/x
\end{aligned} \tag{2}
$$

The basic properties of the above transformation can be summarized as follows: All points (x, y) with $x \neq 0$ are mapped one-to-one to the r-s plane. All points $(0, y)$ with $y \neq 0$ are mapped to infinity. The pencil of directions through the origin, except the y-axis, is mapped to finite points on the s-axis. Figure 2.2.2 illustrates an example of the quadratic transformation T, applied to the singular plane curve in Figure 2.2.1.

Space curves are the intersection of two surfaces and the point singularities of such curves are characterized from one or more of the following situations :

- The gradient of the two surfaces are parallel, e.g., at the origin for surface $z + y^2 - x^3 = 0$ intersected with surface $z + x^2 - y^2 - y^3 = 0$.
- One of the surface gradients is zero, e.g., at the origin for cylinder $y^2 - x^2 - x^3 = 0$ intersected with the plane $z = 0$.
- Both the surface gradients are zero, e.g., at the origin for the cylinder $y^2 - x^2 - x^3 = 0$ intersected with the surface $z^2 - x^2 - y^3 = 0$.

Singular points on space curves are desingularized by transforming the defining surfaces, again using quadratic transformations of the type

$$T:\ x = r$$

$$y = rs$$
$$z = rt \tag{3}$$

This quadratic transformation simplifies the origin, mapping it onto the plane $x = 0$. More complicated singularities are simplified by applying these transformations repeatedly, together with linear transformations.

2.3 Sylvester's Resultant

Consider two homogeneous polynomials in X_1, X_2, with degrees $m > 0$ and $n > 0$ respectively and with coefficients from any euclidean ring[1], i.e., sets which are closed under $+$, $*$, $/$ and elements of which have unique factorizations as well as greatest common divisors.

$$\begin{aligned} F_1(X_1, X_2) &= a_{m,0}{X_1}^m + a_{(m-1),1}{X_1}^{m-1}X_2 + \cdots + a_{0,m}{X_2}^m = 0 \\ F_2(X_1, X_2) &= b_{n,0}{X_1}^n + b_{(n-1),1}{X_1}^{n-1}X_2 + \cdots + b_{0,n}{X_2}^n = 0 \end{aligned} \tag{4}$$

In many situations one needs to know whether or not there are common solutions of F_1 and F_2. For example, F_1 and F_2 might represent two curves in the plane or two surfaces in space and we may interested to know if they have any intersections. One can ascertain the existence of common solutions by computing the Sylvester resultant SR(F_1,F_2) of F_1 and F_2. This SR is *det*(M), the determinant of the $(m+n) \times (m+n)$ matrix M below, derived by multiplying F_1 and F_2 with suitable monomials leading to a linear system of equations in the unknown monomials; it is a polynomial in the *coefficients* of F_1 and F_2, (see also for e.g., [80]). The vanishing of the Sylvester resultant, i.e., SR(F_1,F_2) $= 0$, is both a necessary and sufficient condition that F_1 and F_2 have a common solution.

[1]Example euclidean rings are the rationals $\mathbb{Q}$, the real numbers $\mathbb{R}$, the complex numbers $\mathbb{C}$ and multivariate polynomials defined over them $\mathbb{Q}[\vec{X}]$, $\mathbb{R}[\vec{X}]$, $\mathbb{C}[\vec{X}]$

$$
M = \begin{pmatrix}
a_{m,0} & a_{(m-1),1} & \cdots & a_{0,m} & 0 & 0 & \cdots & 0 \\
0 & a_{m,0} & a_{(m-1),1} & \cdots & a_{0,m} & 0 & \cdots & 0 \\
 & & & & \vdots & & & \\
0 & \cdots & 0 & a_{m,0} & a_{(m-1),1} & & \cdots & a_{0,m} \\
b_{n,0} & b_{(n-1),1} & \cdots & & b_{0,n} & 0 & \cdots & 0 \\
0 & b_{n,0} & b_{(n-1),1} & \cdots & & b_{0,n} & \cdots & 0 \\
 & & & & & \vdots & & \\
0 & & \cdots & 0 & b_{n,0} & b_{(n-1),1} & \cdots & b_{0,n}
\end{pmatrix} \qquad (5)
$$

Efficient symbolic computation of the Sylvester resultant SR(F_1,F_2), has been considered by various authors: for real or complex coefficients see [26, 83] and for multivariate polynomial coefficients see [22, 33].

2.4 Macaulay's Resultant

If $F_1, \ldots, F_n$ are homogeneous polynomials in n variables, then the multivariate resultant MR($F_1, \ldots, F_n$) is a polynomial in the *coefficients* of the F_i that vanishes if and only if all the F_i have a common zero (see for eg., [98]). Geometrically, the multivariate resultant vanishes if and only if the n hypersurfaces ($F_i = 0$) have a common intersection in complex projective space [104].

In deriving the resultant, the F_i are multiplied by suitable monomials to translate the problem of determining whether the polynomials have a common zero, into a problem in linear algebra. A matrix D is constructed whose entries are the coefficients of the $F_1, \ldots, F_n$. However the derived linear equations are not all independent and consequently the determinant of the matrix D properly contains MR. Specifically, Macaulay in [62] shows that $det(D) = det(A) \star$MR where A is an easily derivable submatrix of D. Hence Macaulay's resultant MR is $\frac{det(D)}{det(A)}$.

In evaluating the resultant for particular values of the coefficients, the quotient must first be computed symbolically, treating the coefficients as indeterminates. However, for those values for which the denominator $det(A)$ does not vanish, the symbolic coefficients may be specialized before dividing. Additionally, using differentiation arguments, see [75], one may specialize the coefficients and not encounter the problem of the vanishing denominator.

We now adapt from the method in [62] and show how the matrix D and the submatrix A can be derived for the special case of three homogenous polynomials F_1, F_2, F_3 in three variables

X_1, X_2, and X_3. Let the degrees of F_1, F_2 and F_3 be $m > 0$, $n > 0$ and $p > 0$ respectively. The need for the multivariate resultant for the case of three homogeneous polynomials arises in computing the intersection points of three algebraic surfaces, in intersecting two parametric surfaces, or implicitizing a parametric surface, as we shall see in Sections 3 and 4.

Define an ordering $X_1 < X_2 < X_3$ and imagine a correspondence to exist between the variables X_1, X_2 and X_3 and polynomials F_1, F_2 and F_3 respectively. Let a homogeneous polynomial $F(X_1, X_2, X_3)$, with no monomial $X_1{}^i X_2{}^j X_3{}^k$ divisible by $X_1{}^m$, be said to be reduced in X_1. If further F has no monomials divisible by $X_2{}^n$, it is said to be reduced in X_1 and X_2 and so on. A reduced polynomial F is one which is reduced in two of the three variables.

Now let $d = m + n + p - 2$. Form the homogeneous polynomial $Q = F_1S_1 + F_2S_2 + F_3S_3$ of degree d, where S_1 is a polynomial of degree $n + p - 2$, S_2 is a polynomial of degree $m + p - 2$ reduced in X_1 and S_3 is a polynomial of degree $m + n - 2$ reduced in X_1 and X_2. Construct a square matrix D of the coefficients of F_1, F_2 and F_3 as follows. Matrix D is of size $N \times N$ where the columns correspond to all the N monomials of degree d. It is easily seen that $N = \binom{m+n+p}{2}$. Multiply F_1, F_2 and F_3 in turn with each monomial from S_1, S_2 and S_3 respectively and write the corresponding coefficients under their corresponding monomials, thus giving a row of the matrix. The number of rows is equal to the total number of monomials together in S_1, S_2 and S_3. This number is also equal to N.

Finally, the submatrix A is obtained by deleting columns of D corresponding to monomials reduced in any two of X_1, X_2 or X_3 and the rows corresponding to F_1, F_2 and F_3 for all multipliers reduced in X_2 , X_3 for F_1, and X_1, X_3 for F_2 and X_1, X_2 for F_3. Stating it differently, A consists of the rows and columns which correspond to the non-reduced multipliers and monomials, respectively.

3 Parametric v.s. Implicit Representations

Rationality of the algebraic curve or surface is a restriction where advantages are obtained from having both the implicit and rational parametric representations [41, 45, 87]. While the rational parametric form of representing a curve or surface allows greater ease for transformation and shape control, the implicit form is preferred for testing whether a point is on the given curve or surface and is further conducive to the direct application of algebraic techniques. Simpler algorithms are possible when both representations are available. For example, a straightforward method exists for computing curve - curve and surface - surface intersections when one of the curves, respectively surfaces, is in its implicit form and the other in its parametric form.

3.1 Parameterization

Determining the rational parametric equations of implicitly defined algebraic curves and surfaces, is a process known as parameterization. We present a sketch of the algorithms of degree two and degree three curves and surfaces and give available references to published algorithms for higher degree rational curves and surfaces. All of the described algorithms yield global rational parameterizations in the traditional power basis. However, one may convert them, for instance, to an equivalent Bernstein form by using appropriate power to Bernstein conversion algorithms, see for instance [44].

3.1.1 Plane Curves

The idea of parameterizing a conic is to fix a simple point on the conic and take a one parameter family (pencil) of lines through that point. These intersect the conic in only one additional variable point, yielding a rational parameterization. The intersection of lines through a point on the conic can be efficiently achieved by a linear transformation, mapping the point to infinity along one of the coordinate axis directions. The rational parameterization obtained then is of degree at most 2 and with parameter t corresponding to the slopes of the lines through the point on the conic. Further t ranges from $(-\infty, \infty)$ and covers the entire curve. Details of this algorithm are given in [4].

The idea for parameterizing singular cubics is to take lines through the singular point on the cubic. The actual algorithm is again based on mapping a point on the cubic to infinity, achieved by simple transformations and furthermore the singular point is never explicitly computed. We sketch some details : The general cubic implicit equation is given by

$$C(x,\, y) \;=\; ax^3 \,+\, bx^2y \,+\, cxy^2 \,+\, dy^3 \,+\, ex^2 \,+\, fy^2 \,+\, gxy \,+\, hx \,+\, iy \,+\, j \qquad (6)$$

The y^3 term can be eliminated through a linear coordinate transformation and using another linear substitution the cubic equation can be reduced to the form

$$\bar{y}^2 \;=\; g(\bar{x})\,, \qquad degree\; g(\bar{x}) \;\leq\; 4 \qquad (7)$$

We only need to analyze (7) and see if we can obtain a parametrization for $\bar{x}$ and $\bar{y}$, for then using the earlier linear transformations we can construct the parameterization for x and y. For the case when $g(\bar{x})$ has multiple roots, we do the following. Suppose

$$\bar{y}^2 = \prod_{i=1}^{d} (\bar{x} - \mu_i)^2 \Omega(\bar{x}), \qquad d = 1 \ or \ 2 \tag{8}$$

so each root μ_i occurs an even number of times and $\Omega(\bar{x})$ has no multiple roots. Then if we let

$$\tilde{y} = \frac{\bar{y}}{\prod_{i=1}^{d}(\bar{x} - \mu_i)} \tag{9}$$

equation (7) reduces to

$$\tilde{y}^2 = \Omega(\bar{x}) \tag{10}$$

Note *degree* $\Omega(\bar{x}) \leq 2$, and the above equation is a conic and a rational parametrization is always possible, as shown before. When $g(\bar{x})$ has all distinct roots, the cubic curve can be seen to be nonsingular and hence does not have a rational parametrization. Non-singular cubics are known as elliptic curves or curves of genus 1. However, by solving equation (7), quadratic in $\bar{y}$ a parameterization for the non-singular cubic is obtained of the type that includes a single square root of rational functions. Additional details of this algorithm are given in [5].

A method of computing the genus of algebraic plane curves of arbitrary degree is presented in [6] together with parameterization algorithms applicable for curves of genus = 0. The parameterization techniques, essentially, reduce to solving symbolically systems of homogeneous linear equations and the computation of Sylvester resultants. Tests for the faithfulness of these parameterizations are given in [8, 86].

3.1.2 Space Curves

Algorithms for parameterizations have been given for intersection space curves of two quadric surfaces in [60] where use is made of the fact that a pencil of quadrics contains a ruled surface and in [65] where the algorithm is based on the the computation of eigenvalues of matrices of quadratic forms.

The parameterization algorithms presented in [7] are applicable for irreducible rational space curves C arising from the intersection of two algebraic surfaces of arbitrary degree. The technique presented in [7] is essentailly a method of constructing a plane curve P along with a birational mapping between the points of P and the given space curve C. This together with the results in [6] gives an algorithm to compute the genus of C and if genus $= 0$ the rational parametric equations of C. Together with [8] it also gives a test for the faithfulness of space curve parameterizations. A different method for computing a birational map between a plane curve P and a space curve C as defined before, is also given in [43]. These methods are extended in [15] to compute birational maps between a space curve C, defined by the intersection of two parametric surfaces and a plane curve P in the parametric plane of one of the two surfaces.

3.1.3 Surfaces

The idea of parameterizing quadrics (or conicoids) is identical to the conics. The intersection of lines through a point on the conicoid can again be efficiently achieved by a linear transformation, mapping the point to infinity along one of the coordinate axis directions [4]. This method also straightforwardly generalizes to yield the rational parameterization of conicoid hypersurfaces in arbitrary n-dimensional space.

To construct the rational parametric equations of the cubicoid we need to generate two rational curves (straight lines, conics or singular cubics) on its surface. Note all cubicoids, except the cylinders of nonsingular cubic curves and the cubic cone, are rational. One algorithm for obtaining two different rational curves on the cubicoid is to use the tangent plane intersection method of [5], for two different simple (non-singular) points on the cubicoid. Alternatively, one can generate two non-intersecting straight lines from the twenty seven lines on a cubicoid [88]. All possible configurations as well as the number of real and imaginary straight lines on cubicoids have been accurately classified, see [24, 53, 89].

Let u and v correspond to independent parameterizations of two computed rational curves on a cubicoid. Then the two parameter family (net) of lines defined by two varying points u and v (a variable point u on one rational curve and a variable point v on the other), intersect the cubicoid in one additional point. The equations describing the coordinates of this additional intersection point are the rational parametric equations of the cubicoid. For two non-intersecting lines on the cubicoid with independent parameterization parameters u and v, a point (x, y, z) on the rational cubic surface can be seen to correspond to a single pair (u, v) yielding what is known as a faithful parameterization or a 1-fold covering of the plane. Higher fold coverings are obtained for arbitrary choices of rational curves on the cubicoid.

A method to test for the faithfulness of surface parameterizations is given in [15]. The tangent plane algorithm also generalizes to parameterizing arbitrary degree d hypersurfaces (which are not cylinders or cones), in n-dimensional space for $d \leq n$ [13].

3.2 Implicitization

We now present algorithms for determining the implicit equation of parametrically defined algebraic curves and surfaces, a process known as implicitization. The implicitization techniques described here are general, and apply to arbitrary degree rational curves and surfaces.

3.2.1 Plane Curves

The implicitization of parametrically defined plane algebraic curves is achieved by eliminating the parameter from the two parametric equations, see [36, 45, 76, 87]. More specifically, consider the rational parametric representation of an algebraic plane curve:

$$\begin{aligned} x &= \frac{f_1(t)}{f_3(t)} \\ y &= \frac{f_2(t)}{f_3(t)} \end{aligned}$$

Here f_1, f_2 and f_3 are polynomials in t. To find the implicit equation $h(x, y) = 0$ corresponding to the above two equations we first homogenize the polynomials $f_1(t)$, $f_2(t)$, $f_3(t)$ with a homogenizing parameter w to yield homogeneous polynomials $\bar{f}_1(t, w)$, $\bar{f}_2(t, w)$, $\bar{f}_3(t, w)$ and then construct polynomial equations

$$\begin{aligned} F_1(t, w) &= \bar{f}_3(t, w)x - \bar{f}_1(t, w) = 0 \\ F_2(t, w) &= \bar{f}_3(t, w)y - \bar{f}_2(t, w) = 0 \end{aligned}$$

Then the implicit equation $h(x, y) = \mathrm{SR}(F_1, F_2)$. To see why, remember that $\mathrm{SR}(F_1, F_2) = 0$ if and only if $F_1(t, w) = 0$ and $F_2(t, w) = 0$ have common t and w solutions. Note then that whenever the implicit equation $h(x, y) = 0$, there is a value for the parameter t that simultaneously satisfies the parametric equations for x and y. The coefficients of polynomials F_1 and F_2 from which the curve implicit equation is derived, are special — namely, they are at most linear in x and y. If the degree of F_1 and F_2 in t is d_1 and d_2, the degree of the implicit equation in x and y is at most degree $Max(d_1, d_2)$. This from the known degree bound, see for e.g. [80], where the coefficients of F_1 (respectively F_2) appear in SR with the degree of F_2 (respectively, degree of F_1).

One addtional fact to rememeber in using the Sylvester resultant (a homogeneous projection), for the affine operation of implicitization, is that one needs to divide out in certain cases by the extraneous component at infinity. See [10] for details.

3.2.2 Surfaces

The implicitization of parametrically defined algebraic surfaces requires the simultaneous elimination of two parameter variables from the three parametric equations. Eliminating two variables from three equations by taking the Sylvester resultant of groups of two polynomials leads to extraneous factors. In practice, this means that the resulting implicit form, describes not only the parametric surface, but in addition, other surfaces. Various simultaneous eliminants of many variables, known as multivariate resultants, have also been defined in the literature: by Cayley [80], by Hurwirtz, see[98], and by Macaulay [62] as explained in Section 2.5.

More specifically, consider the rational parametric representation of an algebraic surface:

$$\begin{aligned} x &= \frac{f_1(s,t)}{f_4(s,t)} \\ y &= \frac{f_2(s,t)}{f_4(s,t)} \\ z &= \frac{f_3(s,t)}{f_4(s,t)} \end{aligned}$$

Here f_1, f_2, f_3 and f_4 are polynomials in s, t. To find the implicit equation $h(x,y,z) = 0$ corresponding to the above three equations, we first homogenize the polynomials $f_1(s,t)$, $f_2(s,t)$, $f_3(s,t)$, $f_4(s,t)$ with a homogenizing parameter w to yield homogeneous polynomials $\bar{f}_1(s,t,w)$, $\bar{f}_2(s,t,w)$, $\bar{f}_3(s,t,w)$, $\bar{f}_3(s,t,w)$ and then construct polynomial equations

$$\begin{aligned} F_1(s,t,w) &= \bar{f}_4(s,t,w)x - \bar{f}_1(s,t,w) = 0 \\ F_2(s,t,w) &= \bar{f}_4(s,t,w)y - \bar{f}_2(s,t,w) = 0 \\ F_3(s,t,w) &= \bar{f}_4(s,t,w)z - \bar{f}_3(s,t,w) = 0 \end{aligned} \qquad (11)$$

Then the implicit equation $h(x,y,z) = \mathrm{MR}(F_1, F_2, F_3)$. To see why, remember that $\mathrm{MR}(F_1, F_2, F_3) = 0$ if and only if $F_1(s,t,w) = 0$, $F_2(s,t,w) = 0$ and $F_3(s,t,w) = 0$ have common s, t and w solutions. Next, note that whenever the implicit equation $h(x,y,z) = 0$, there is a value for the parameters s and t that simultaneously satisfies the parametric equations for x, y and z. If the degrees of polynomials F_1, F_2 and F_3 in s and t are d_1, d_2 and d_3 respectively, then the degree of the implicit equation in x, y and z is at most degree $Max(d_1d_2, d_1d_3, d_2d_3)$. This from the known degree bound, see for e.g. [62], where the coefficients of F_1 (respectively F_2 and F_3) appear in MR with the product of the degrees of F_2 and F_3 (respectively, product of degrees of F_1 and F_3, product of degrees of F_1 and F_2).

There is then again the problem of the extraneous component at infinity when using it for implicitizing. Similar to the Sylvester resultant, the Macaulay multivariate resultant is homogeneous and for the affine operation of implicitization, requires the elimination of the projected component at infinity [10]. In the affine case it is useful to circumvent the division step for the multivariate resultant by the use of a single variable perturbation of the highest degree terms of the parametric equations and the computation of characteristic polynomials [29].

4 Solutions of Systems of Polynomial Equations

Numerical operations on geometric models with algebraic surface boundaries reduce to computing the solutions of various instances of systems of polynomial equations. We shall also encounter this in the geometry processing operations of Sections 5 to 9. Here we consider the relevant special cases of the general problem, sketching solution procedures using the polynomial resultants of Sections 2.3 and 2.4. Analogous algorithms can also be devised using the elimination (triangulation) procedure of the Gröbner basis computation under appropriate variable orderings, see [27].

4.1 Polynomials in One Variable

1. Real and Complex Roots

$$f(x) = 0$$

The fundamental problem of computer algebra, as it is often referred to, is that of determining all the real and complex roots of a polynomial equation $f(x) = 0$. Several solutions have been offered over the years, some restricted to only the determination of real roots. Both numerical and symbolic techniques exist, where the goal is to determine the solutions to within ϵ-approximations of the true solutions. Stable numerical techniques for determining the real and complex roots of f with real coefficients is presented in [54] and the roots of f with complex coefficients in [55]. Symbolic root isolation techniques using Sturm sequences for the real roots are given in [84] while [74] also considers the case of complex roots.

- Common Roots ?

$$f_1(x) = 0$$

$$f_2(x) = 0$$

Common roots of f_1 and f_2 exist iff $\mathrm{SR}(f_1, f_2)$ vanishes identically. The common roots themselves are computed from the polynomial $GCD(f_1, f_2)$. Remember, GCD is the Greatest Common Divisior. For GCD computations see for e.g. [26, 84].

- Common Roots ??

$$f_1(x) = 0$$

$$f_2(x) = 0$$

$$f_3(x) = 0$$

To check for the existence of common roots of f_1, f_2 and f_3 one may use the Kronecker method of indeterminates, see [98]. On choosing indeterminates u_1, u_2, u_3 and v_1, v_2, v_3 one sees that common roots of the above three polynomials exist iff $\mathrm{SR}(u_1 f_1 + u_2 f_2 + u_3 f_3, v_1 f_1 + v_2 f_2 + v_3 f_3)$ vanishes identically. Again the common roots can be obtained from polynomial GCD computations.

4.2 Polynomials in Two Variables

- Display Algebraic Curves

$$f(x, y) = 0$$

Here by a solution is meant a correct topological trace of prespecified unambiguous real parts of the algebraic plane curve $f(x, y) = 0$. At times one may also desire a trace of all closed real components of f within a specified region of the plane. In full generality, the robust tracing of algebraic plane curves is a difficult problem. In [69], Pratt and Geisow review several such methods. A common problem stems from the inherent geometric complexity of singularities of high degree algebraic curves. In particular, such a curve may possess singular points where the curve has an abrupt change of normal direction (cusps), multiple self-intersecting branches (nodes), or self-tangent branches (tacnodes).

One solution which applies for the case of rational (genus = 0) algebraic curves is to first construct the rational parametric equations of the curve, cf., Section 3.1.1. Tracing plane curves which are given parametrically, then simply amounts to evaluating the parametric equations for several distinct parameter values. Appropriate choices of distinct parameter values can be made to yield a curvature dependent tracing of the curve, suitable for graphics [66]. See also Figure 2.1.1 and 2.1.2, where such a tracing procedure was adopted.

For the case of implicit (non-rational) algebraic curves, [44, 79] proposes subdivision. Briefly, the curve $f(x, y) = 0$ is conceptualized as the intersection of $z = f(x, y)$ and $z = 0$, and after translating $z = f(x, y)$ into Bernstein form, several subdivision schemes are proposed for evaluating the curve in small regions in which it is well behaved. Although the method can cope with many singularities, no analysis is made to identify branch connectivity or to give an analysis of the structure of the singularity. In [37], power series are constructed to locally approximate plane algebraic curves and surface intersections. The method technically relies on the Implicit Function Theorem, seeking to represent a curve branch explicitly in one coordinate as function of the other coordinate(s). The advantage of such a representation, is that it allows simple stepping techniques. On the other hand, the quality of approximation is limited by a more stringent convergence criterion, and the method does not handle singular points. See [12] where extensions to compute power series expansions at singular points are presented.

There are also algorithms for analyzing the topology of real algebraic curves in the plane, e.g., [9]. Based on cylindrical algebraic decomposition, [34, 84], these algorithms make extensive use of symbolic computation and root isolation to locate *critical* curve points, that is, singularities and points whose tangents are parallel to one of the coordinate axes. Thereafter, the critical points are connected with curve segments that are simple to trace. There however remains the non-trivial task of assuring correct branch connectivity at the critical singular points having various intersecting and tangential branches, a problem we now address.

For plane algebraic curves, [17] shows that correct branch connectivity can always be achieved by utilizing results from Section 2.2. The trace of $f(x, y) = 0$ commences at a given input point with a desired direction. At noncritical segments, one proceeds numerically, using a scheme in which the curve is locally approximated by a low degree Taylor polynomial interpolant and a new curve point estimate is derived from it by taking steps of variable lengths. Newton iteration is then used to refine this new point estimate. When the condition number of the system becomes very large, one tries to locate a nearby curve singularity. Then, by applying quadratic transformations of Section 2.2, the branch of f to be traced is birationally mapped to a branch of a transformed curve g that has no singularities. The transformed branch is traced and the points of g are mapped to corresponding points of f. The trace of g continues until we have passed the singularity of f. In this way, correct branch connectivity is achieved. See again the Figures 2.2.1 and 2.2.2, which were traced using this procedure.

- Intersection Points of Curves

$$f_1(x,y) = 0$$

$$f_2(x,y) = 0$$

A number of alternative ways exist for computing the finite number of point intersections of two plane curves [36, 75]. The number of intersection points is bounded by the product of the degrees of the polynomials f_1 and f_2. The most promising is the method of computing birational maps using the Sylvester resultant of Section 2.3. Here one of the two variables, say y, is eliminated (birationally) using $\mathrm{SR}(f_1, f_2)$. This yields a polynomial $p(x)$. Further, the eliminated variable y can be expressed as rational functions of the variable x. Hence, computing the roots of a univariate polynomial $p(x)$, in turn, efficiently yields the coordinates of the intersection points.

For the case where one of the two curves is rational, we may alternatively intersect them by first constructing the rational parametric equations of the rational curve, cf., Section 3.1.1. Substituting these parametric equations into the implicit equation of the other curve yields a polynomial in a single variable, the roots of which again give the coordinates of the intersection points.

Finally, we note that there may be an entire common curve component, a case of excess intersection. This occurs iff $\mathrm{SR}(f_1, f_2)$ vanishes identically. The common curve component can be recovered by computing the $GCD(f_1, f_2)$.

- Curves Intersect ?

$$f_1(x,y) = 0$$

$$f_2(x,y) = 0$$

$$f_3(x,y) = 0$$

Common points of intersection of three plane curves exist iff $\mathrm{MR}(f_1, f_2, f_3)$ vanishes identically. Alternatively, one may use the Kronecker method of indeterminates and Sylvester resultants. On choosing indeterminates u_1, u_2, u_3 and v_1, v_2, v_3 one sees that common roots of the above three polynomials exist iff $\mathrm{SR}(u_1 f_1 + u_2 f_2 + u_3 f_3, v_1 f_1 + v_2 f_2 + v_3 f_3)$ vanishes identically. The common intersection points can be obtained by computing the common roots of any two of the above three equations. Next one verifies which of those roots also satisfy the remaining third equation.

There may also be an entire common curve component, a case of excess intersection. This occurs iff the $GCD(f_1, f_2, f_3)$ is non-trivial.

4.3 Polynomials in Three Variables

- Display Surfaces

$$f(x, y, z) = 0$$

Methods have been proposed here in computer graphics literature for the display of real implicit surfaces . These include techniques for ray tracing, and surface polygonalization or meshing, see for e.g. [64].

For rational implicit surfaces an alternative is to first construct a rational parameterization of $f(x, y, z)$, as sketched in Section 3.1.3. The algorithms essentially require either one or two simple points on the algebraic surface, to construct a rational parameterization. Generating many more points on the algebraic surface then simply amounts to evaluating the parametric equations for several distinct parameter values. See the Figures 2.1.3 and 2.1.4 which were displayed using this procedure. It is an interesting open problem to appropriately choose distinct parameter values to yield curvature dependent plots of the surface, suitable for graphics.

- Intersection Curve of Algebraic Surfaces

$$f_1(x, y, z) = 0$$

$$f_2(x, y, z) = 0$$

Here again, by a solution is meant a correct topological trace of prespecified unambiguous parts of the real intersection curve of two implicitly defined algebraic surfaces. Also at times one may desire a trace of all closed real components of the real curve, within a specified region of space. This approach to surface intersection curve tracing applies directly to boolean set operations on algebraic boundary models. Moreover, when rendering curved faces, silhouette curves need to be determined and these are the intersection of two surfaces.

It is worth noting that a tracing procedure for plane algebraic curves can yield a tracing procedure for the intersection curve of algebraic surfaces.

1. When the faces of a model are parametric patches, with a-priori known implicit equations, edges bounding these patches can be represented as plane curves in the parameter plane of one of the faces. Here, the surface implicitization techniques of Section 3.2.2 prove useful.

2. When intersecting two implicit surfaces $f_1(x,y,z)=0$ and $f_2(x,y,z)=0$, one of them, say f_1, might possess a rational parameterization. If so, the parametric form can be determined in certain cases, see Section 3.1.2. By substituting thereafter the parametric equations of f_1 into the implicit equation of f_2, a plane curve in the parameter plane is obtained that is in birational correspondence with the intersection curve of f_1 and f_2. Here, birational correspondence means that in each direction rational maps exist.

3. When intersecting nonrational implicit surfaces $f_1(x,y,z)=0$ and $f_2(x,y,z)=0$, one can always find a rational surface $f_3(x,y,z)=0$ containing the intersection curve of f_1 and f_2. This, by using the Sylvester resultant of Section 2.3, $\mathrm{SR}(f_1,f_2)$, where one can construct a birational map to a projected plane algebraic curve and which, as part of the map so constructed, determines f_3, see [7, 43].

Note, however, that in all of the above cases the corresponding plane curve might have more singularities than the space curve. Moreover, the degree of the curve is the product of the surface degrees, so that tracing the corresponding planar curve is numerically more delicate. If the birational map is not derived carefully, finally, the degree of the plane curve may be even higher. Thus, for simple singularities, a numerical approach remains attractive.

The numerical part of the tracing procedure for plane curves of Section 4.2 can also be generalized directly to the tracing of the intersection curves of two algebraic surfaces. The intersecting surfaces $f_1(x,y,z)=0$ and $f_2(x,y,z)=0$, can in fact be arbitrary smooth functions. A strength of the method lies in its ability to consolidate the computation needed for the Newton iteration with the computation determining the power series expansion. Moreover, there is a close correspondence of the computational machinery needed by the method with an algebraic procedure for analyzing the curve at singular points. This correspondence permits it to cope directly with a large class of space curve singularities. An example of such a trace is shown in Figure 4.3.1. Details can be found in [17].

Finally, we note that there may also be an entire common surface component, a case of excess intersection. This occurs iff $\mathrm{SR}(f_1, f_2)$ vanishes identically. The common surface component can be recovered by computing the $GCD(f_1, f_2)$.

- Intersection Points of Algebraic Surfaces

$$f_1(x, y, z) = 0$$

$$f_2(x, y, z) = 0$$

$$f_3(x, y, z) = 0$$

Solutions of different dimensions may arise, viz., points (dimension 0), curves (dimension 1) and surfaces (dimension 2).

For the case where the solutions are all points, the number of intersections are finite and bounded by the product of the degrees of the polynomials f_1, f_2 and f_3. The method to be used here is popularly known as the U-resultant technique [98]. A linear equation $f_4 = u_1x + u_2y + u_3z$ is additionally taken, involving new indeterminates u_1, u_2, u_3. When common intersection points exist, the polynomial $\mathrm{MR}(f_1, f_2, f_3, f_4)$, (of Section 2.4), in the new indeterminates, decomposes into linear factors from which the coordinates of the solution points can again be reconstructed, see [28, 29, 58, 75].

There may also be an entire common curve component in the solution, a case of excess intersection. This occurs iff $\mathrm{MR}(f_1, f_2, f_3, f_4)$, eliminating the three variables (x, y, z), vanishes identically. The common curve component can be recovered by computing a birational planar projection of the space curve using $\mathrm{SR}(u_1f_1 + u_2f_2 + u_3f_3, v_1f_1 + v_2f_2 + v_3f_3)$ and additionally expressing the eliminated variable as rational functions of the plane curve variables. Details of computing such birational maps can be found in [14].

There may also be an entire common surface component, a worse case of excess intersection. This occurs iff the $GCD(f_1, f_2, f_3)$ is non-trivial.

4.4 Polynomials in Four Variables

- Display Hypersurfaces in 4D

$$f(x, y, z, w) = 0$$

For rational implicit hypersurface surfaces (for example when its degree is less than or equal to four) an alternative is to first construct a rational parameterization of $f(x, y, z, w)$, see for e.g., [13]. The parameterization algorithm essentially requires computing some simple points on the hypersurface. Generating many more points on the algebraic hypersurface surface then simply amounts to evaluating the parametric equations for several distinct parameter values. Additionally for general hypersurfaces one may use generalized methods of ray tracing and hypersurface triangulation.

- Intersection Surfaces in 4D

$$f_1(x, y, z, w) = 0$$

$$f_2(x, y, z, w) = 0$$

Using the Sylvester resultant SR of Section 2.3 to eliminate w, the common solution of dimension 2 of f_1 and f_2 can be represented by the projected surface in (x, y, z) space, together with w equal to a rational function in x, y, z, which maps points on the surface to points on the hypersurface. Explicit points on the hypersurface can then be computed using the rational map together with algorithms of Section 4.3 for computing points on surfaces. Techniques are presented in [14] to construct these rational maps.

There may also be an entire common hypersurface component (dimension 3), a case of excess intersection. This occurs iff $\mathrm{SR}(f_1, f_2)$ vanishes identically. The common surface component can be recovered by computing the $GCD(f_1, f_2)$.

- Intersection Curve of Parametric Surfaces

$$f_1(x, y, z, w) = 0$$

$$f_2(x, y, z, w) = 0$$

$$f_3(x, y, z, w) = 0$$

The solution for this case is again a trace of the real intersection curve of two parametric surfaces given by $(x = G_{1,1}(u_1, v_1), y = G_{2,1}(u_1, v_1), z = G_{3,1}(u_1, v_1))$ and $(x = G_{1,2}(u_2, v_2), y = G_{2,2}(u_2, v_2), z = G_{3,2}(u_2, v_2))$, where the $G_{i,j}$, $i = 1, 2, 3$, $j = 1, 2$, are rational functions. The intersection is defined by $f_j(u_1, v_1, u_2, v_2) = G_{j,1}(u_1, v_1) - G_{j,2}(u_2, v_2)$, $j = 1, 2, 3$, a system of three equations in four unknowns. The tracing procedure is exactly the same as the one in Section 4.2, with details in [17].

Using the Macaulay resultant of Section 2.4, $MR(f_1, f_2, f_3)$, one may also construct a birational map to a plane algebraic curve, which is the projection of the space curve into the parametric plane of either of the two parametric surfaces [15].

5 Solid Model Representations

5.1 Algebraic Boundary Model

By far the most natural and comprehensive representation of solids enclosing finite volume is by an enumeration of its finite number of boundary components. The boundary geometric elements consist of point vertices, curved edges and bounded surface patches. This, together with the capability of boolean set and other geometric solid operations of Sections 6 to 9, generalize both traditional B-rep and CSG based models into a single hybrid representation scheme, see [71, 101]. In specialized schemes some solids may be chosen as primitives and stored with explicit parameterized boundaries, instantiated by certain select parameters.

A comprehensive boundary representation of an object with general algebraic surfaces, thus consists of the following:

- A finite set of vertices usually specified by Cartesian coordinates.
- A finite set of directed edges, where each edge is incident to two vertices. Typically, an edge is specified by the intersection of two faces, one on the left and one on the right. Here left and right are defined relative to the edge direction as seen from the exterior of the object. Further an interior point is also provided on each edge which helps remove any geometric ambiguity in the representation for high degree algebraic curves, [71]. Geometric disambiguation may also be achieved by adding tangent and higher derivative information at singular vertices, [51].
- A finite set of faces, where each face is bounded by a single directed cycle of edges. Each face also has a surface equation, represented either in implicit or in parametric form. The surface equation has been chosen such that the gradient vector points to the exterior of the object.

In addition edge and face adjacency information are provided in the form of cyclically sorted edges about a vertex and faces about an edge. Additional conventional assumptions, for non-ambiguity and non-degeneracy, are also made, e.g., edges and faces are non-singular, two distinct faces intersect only in edges, an auxiliary surface is specified for each edge where adjacent faces meet tangentially, etc. We restrict our representations to solids which enclose non-zero, finite volume. Hence non-regularities such as dangling edges and dangling faces which depending on one's viewpoint enclose zero or infinite volume, are not permitted.

A *planar* geometric model with algebraic boundary curves has a similar though specialized boundary representation consisting of:

A single simple directed cycle of algebraic curve edges, where each edge is directed and incident to two vertices. Each edge also has curve equations, which are implicit and/or rational parametric. Further an interior point is also provided on each implicitly defined edge which helps remove any geometric ambiguity in the case of vertices which are singularities of the algebraic curve. Finally, each vertex is exactly specified by Cartesian coordinates.

The curve equations for each edge are chosen such that the direction of the normal at each point of the edge is towards the exterior of the object. For a simple point on the curve, the normal is defined as the vector of partials to the curve evaluated at that point. For a singular point on the curve we associate a range of normal directions, determined by normals to the finite number of tangents at the singular point. Finally, the orientation of the cycle of edges is such that the interior of the object is to the left when the edges are traversed.

5.2 Gaussian Models

For particular geometric model operations such as sweep and convolution it is very useful to have an additional, alternate model representation. For the restricted class of convex solid objects this representation, variously known as the Gaussian model [49], explicitly captures the piecewise tangent space description of the solid's boundary elements.

Let S^2 be the unit sphere in R^3, and $Bdr(T)$ be the boundary of a convex set $T \subset R^3$. For any set $K \subset Bdr(T)$, we shall define a set $N(T,K) \subset S^2$ as follows. A point $e \in S^2$ belongs to $N(T,K)$ if there exists a point $p \in K$ and a tangent plane L_p at p such that e is the exterior normal to L_p. This set $N(T,K)$ is called the *Gaussian Image* of K. The function $N(T,\cdot) : P(Bdr(T)) \rightarrow P(S^2)$ is called the *Gaussian Map* of T, where $P(Bdr(T))$ and $P(S^2)$ are the power sets of $Bdr(T)$ and S^2. It is a bijective map and its inverse $N^{-1}(T,\cdot) : P(S^2) \rightarrow P(Bdr(T))$ is called the *Inverse Gaussian Map* of T. For any set $G \subset S^2$, the *Inverse Gaussian Image* of G is defined by $N^{-1}(T,G)$. The

Gaussian Curvature of $p \in Bdr(T)$ is the limit of the ratio (Area of $N(T, K)$) / (Area of K) as K shrinks to the point p, see [67]. Gaussian curvature at a point on a surface is also the product of the two principal curvatures at that point [49].

Now a convex, algebraic boundary model is one which consists of boundary surface patches, where at all points on each patch, the Gaussian curvature is ≥ 0. Additionally, the set of all points enclosed by the model, is a convex set. The Gaussian Image of such a convex, algebraic boundary model covers S^2 completely and partitions S^2 into a set of generic faces (surface patches) as described below. Certain generic faces on S^2 degenerate to curves and points. Using the adjacency graph of vertices, edges and faces of the boundary model, the generic faces of S^2 are connected with the same topology. In particular the Gaussian Model of a convex algebraic boundary model consists of a finite set of vertices, edges and faces on the surface of a unit sphere S^2 as follows:

- Consider first, the boundary faces. A face is an algebraic surface patch which is either elliptic (*Gaussian Curvature* is positive at each point), ruled or planar (*Gaussian Curvature* is zero at each point). For an elliptic face F, the Gaussian Image $N(T,F)$ is a patch of S^2 with its boundary curves determined by the normals to the tangent planes of F at the boundary. That is, the boundary of $N(T,F)$ consists of the set of points $\frac{\nabla f(p)}{\|\nabla f(p)\|}$ for $p \in \cup_{E \in \Gamma} E$, where $f = 0$ is the surface equation of F and Γ is the set of boundary edges of F. For a ruled surface face F, $N(T,F)$ is a degenerate curve on S^2 and for a planar face F, $N(T,F)$ is a degenerate point on S^2.

- Consider next, an edge E defined by two intersecting faces F and G, where F and G meet either transversally or tangentially along E. When F and G meet transversally along E, each point $p \in E$ determines two different points n_F and n_G on S^2 determined by the exterior normals of the tangent planes of F and G at p. Then $N(T,p)$ is the geodesic arc (part of a great circle) γ_p connecting n_F and n_G on S^2 and $N(T,E) = \cup_{p \in E} \gamma_p$ is a patch of S^2. The set $N(T,E)$ has 4 boundary curves given by the set of points $\frac{\nabla f(p)}{\|\nabla f(p)\|}$ for $p \in E$, the set of points $\frac{\nabla g(p)}{\|\nabla g(p)\|}$ for $p \in E$, and the geodesic arcs γ_{p_S} and γ_{p_E}, where $f = 0$ and $g = 0$ are the surface equations of F and G, and p_S and p_E are the starting and ending vertices of E. When F and G meet tangentially along E, $N(T,E)$ is a degenerate curve on S^2. In particular, $N(T,E)$ is the common boundary curve of $N(T,F)$ and $N(T,G)$. That is, it is the set of points $\frac{\nabla f(p)}{\|\nabla f(p)\|} = \frac{\nabla g(p)}{\|\nabla g(p)\|}$ for $p \in E$. When F and G are planar faces, E is a linear edge and $N(T,E)$ is a degenerate geodesic arc γ connecting n_F and n_G on S^2, where n_F and n_G are the exterior normals of F and G.

- Consider finally, a vertex p defined by k adjacent faces $F_1, F_2, \ldots, F_k$ intersecting at p (ordered via their normals at p in a counter–clockwise direction). Each face F_i corresponds to a point n_i on S^2 determined by the normal of F_i at p. Let γ_i ($i = 1, \ldots, k$) be the geodesic arc on S^2 connecting n_i and n_{i+1} where $n_{k+1} = n_1$. Then $N(T, p)$ is the convex patch on S^2 bounded by the cycle of geodesic arcs $\gamma_1, \gamma_2, \ldots, \gamma_k$. When F_i and F_{i+1} are tangent at p, γ_i is a degenerate point $n_i = n_{i+1}$. In the special case of all k faces being tangent at p, the entire set $N(T, p)$ is a degenerate point. The set $N(T, p)$ can also be a degenerate geodesic arc on S^2 when $Bdr(T)$ is locally smooth at p except along a curve which is tangent at p.

Figure 5.2.2 shows the Gaussian Model for the convex object in Fig. 5.2.1. In Figure 5.2.1, face F_3 is a ruled surface and face F_2 is a planar patch. The corresponding Gaussian Images degenerate into a curve and a point respectively. Further since faces F_1 and F_3 are tangent to each other along E_2, the Gaussian Image of E_2 also degenerates into a curve.

5.3 Constructive Semi-Algebraic Models

In certain applications, an implicit representation of solids may also be used, consisting essentially of finite boolean combinations of geometric elements defined by algebraic inequalities or semi-algebraic sets of arbitrary degree. Contrary to explicit boundary representations, the solids are modeled here together with the volume they enclose. The enclosed volume is allowed to be both finite or infinite, with the allowed geometric elements themselves also exhibiting both properties. In special cases, simplifying assumptions in the type of allowed geometric elements may also be made, for e.g., the algebraic boundary of the semi-algebraic sets may be assumed to be smooth, connected, not of mixed dimensions, etc. See [11, 28, 34, 71, 84] for some details of the various properties and operations applicable to such general representations. Representations of this type are very compact and simple. However, due to inherent time-intensive computation requirements, they have not yet made an impact on geometric modeling practices. Whether specialized versions will eventually be competitive in space or time remains to be seen.

6 Boolean Operations on Models

The ability to compute the union, intersection and difference of two geometric models and yield a third model with the same representation, provides a geometric modeling system at once with a sophisticated way of synthesing and creating models of physical objects with various complex geometries. The same boolean operations also provide a way of doing static interference checks to confirm if two different models occupy a common region in space.

It is quite simple to observe that boolean operations cause dangling edges and faces and hence destroy the regularity of input model representations. Modified boolean operations, i.e. regularized union, intersection and difference, are thus used instead with the property of preserving regularity [73]. Also the operation of complementation, especially for boundary representations, which reduces simply to an orientation change of directed face-edge cycles, causes one to concentrate only on the boolean set operation of intersection, with the operations of union and difference being then defined in terms of intersection and complementation using De Morgan's laws.

The intersection of two algebraic boundary models consists of two major algorithmic subparts. There are the extensive numeric calculations of pairwise boundary curve and surface intersections or incidences. This is followed by a containment classification for boundary vertices, edges and faces and a topological reconstruction of the resulting solid's boundary representation. These subparts are prone to errors in numerical calculation, [42], as well as errors in topological decisions based on approximate numerical solutions, [52, 95]. A host of robust algorithmic paradigms need to be incorporated before efficient and error-free boolean operations become possible. See [57] for a sample approach in the domain of polyhedral models.

We now detail some of the intersection operations for algebraic boundary models required for each of the two algorithmic subparts discussed above.

6.1 Intersections, Incidences and Containment Classifications

A wide spectrum of intersection and incidence tests with algebraic curves and surfaces need to be accomplished, as listed below. In particular, one needs to compute the intersection of:

- two surfaces, defined implicitly or parametrically .

- three surfaces, which is also equivalent to the simultaneous intersection a space curve with a surface, with surfaces defined parametrically or implicitly.
- four surfaces, which is also equivalent to the simultaneous intersection of two space curves, both defined as the intersection of implicit or parametric surfaces.

The numeric computation of these tasks trivially reduce to the computation of solutions of systems of polynomial equations of Section 4.

The boundary elements of the model are edges and faces which are pieces of algebraic curves and surfaces, respectively. To decide whether intersections occur between edges and faces, as well as for later topological decisions, one additionally requires "in/out" classification tests. Specifically, the containment of :

- a vertex "in/out" of edges and faces.
- an edge "in/out" of faces.

The "in/out" containments are various applications of a sorting procedure for points along an algebraic curve, see [56]. The algorithms of many of these classification tests also reduce to the sorting of :

- edges about a vertex.
- faces about an edge.

6.2 Boundary Topology Reconstruction

One of the main subtasks is the determination of the nesting structure of face loops, consisting of a closed chain of piecewise curves on a surface . These loops arise from the intersections between the element curves and surfaces of the intersecting solids, as detailed in the previous subsection. Knowing the nesting structure, i.e., the inner and outer intersection curve loops allows for the correct reconstruction of the resulting solid. Algorithmic solutions for this problem consist of a generalized sweep of a curve on a surface, see for e.g., [103]. For the case of rational surfaces, a planar sweep in the parametric plane, also sufficies. For algorithms using planar sweeps, see for e.g., [30, 39].

For robust computations which always yield consistent boundary topologies, one needs to make specific topological decisions based on imprecise numerical data, [52, 95]. The methodology we adopt is to live with uncertainity. Namely, the choices that ϵ is negative, zero or positive, are equally likely. Decision points, where several choices may exist, are to be considered either "independent" or "dependent". At independent decision points, any choice may be made from the finite set of possibilities while the choice at dependent decisions points ensures the invariant state of global consistency. This consistency, for now, is achieved by means of topological reasoning and is specific to each problem and its desired goals.

For these topological consistency checks, we require the minimum distances between pairs of vertices, edges and faces of the model, to decide degenerate incidences. Specifically, one needs to compute minimum distances between:

- a vertex and an edge, or a vertex and a face
- two edges
- two faces

A significant time is also spent in the analysis of intersection curve singularities, see for e.g., [3, 17, 73].

7 Decomposition Operations on Models

The main purpose behind decomposition operations is to simplify a problem for models with complex geometries into a number of subproblems dealing with models having simple boundaries. Simplifications are possible for instance in geometric point location and intersection detection problems. In most cases a decomposition, in terms of a finite union of disjoint convex pieces is useful and this is always possible for polyhedral models, see for e.g., [30]. However a similar decomposition is in general, impossible for curved models, not even for simple curved models in the plane. Annular disks or toruses, serve as examples of this complication. Additional complexities in decompositions for the curved world also arise from the nature of the boundary curves and surfaces. It is possible, although non-trivial, to decompose an algebraic surface into elliptic surface patches (all points having Gaussian curvature > 0), hyperbolic surface patches (all points having Gaussian curvature < 0) and parabolic curves (all points having Gaussian curvature $= 0$) [49]. This, in analogy with algebraic curves, which can always be decomposed into convex curve segments (curvature > 0), concave curve segments (curvature < 0) and flex points (curvature $= 0$). See for example, [21, 56] where procedures to achieve such a curve decomposition, are presented.

For curved models with algebraic surface boundaries, a number of alternative decompositions have been proposed. It is possible to decompose algebraic boundary models into the disjoint union of certain primitive pieces. In the case of the "cylindrical algebraic decomposition" of [34], the primitive pieces are cells over which a set of real polynomials, (derived from a boolean formula of polynomial inequalities), has constant sign. In the proposed "funnel decomposition" of [31], the decomposition is coarser, and each cylindrical cell of the decomposition has constant spatial description. The latter decomposition is geometrically motivated, where individual cells possess monotonicity and vertical visibility properties, as opposed to the former decomposition of [34], which is algebraic, where individual cells possess algebraic sign invariance properties. There also exist closest point "Voronoi decompositions" and "Whitney stratifications" of algebraic boundary models [28], which similar to [31, 34], decompose the entire ambient three dimensional space along with the models.

The funnel decomposition is a three dimensional generalization of the horizontal-vertex-visibility decomposition of a simple polygon in the plane [96]. In [38] the horizontal-vertex-visibility decomposition is generalized to a visibility decomposition of a planar curved model. See Figure 7.1 for an example. This visibility decomposition, next allows a simple decomposition of the curved model into a union of monotone pieces and also into a union of differences of unions of possibly overlapping convex curved pieces. In [21] a different algorithm is presented, to construct a simple *characteristic* carrier polygon of planar geometric model. A *characteristic* carrier polygon is a simple polygon which differs from the original object by convex regions each of which is either totally contained in the interior of the object or in its exterior. See Figure 7.2. By refining this carrier polygon further, one is able to construct an *inner* polygon (resp. an *outer* polygon) which is a simple polygon totally contained in (resp. totally containing) the geometric model. Using the simple inner, outer and characteristic polygons, the following can be computed (1) a convex decomposition of the geometric model as a difference of unions of disjoint convex models, (2) a decomposition of the geometric model as a union of disjoint certain primitive models.

A number of subtasks, requiring techniques from Section 4, need to be performed to achieve such planar model decompositions. These include:

1. The intersection of a line and a curve segment.

2. The computation of the tangent line to a curve segment from a point .

3. The computation of the common tangent line between two curve segments.

For three dimensional decompositions we additionally need:

1. The intersection of two surfaces

2. The projection of intersection space curves, to form the base curves of cylinders

3. The computation of singular and extremal, points/curves of surfaces

8 Convex Hull Operations on Models

The convex hull computation is a fundamental one in computational geometry. There are numerous applications in which the convex hulls of complex models can be used effectively to make certain geometric decisions easier. For example, a null intersection between the convex hulls of two models implies a null intersection between the original models. Since intersection testing for convex models is easier than for non-convex models, convex hulls intersection is used as an efficient first test in a general object intersection-detection algorithm. Additional motivation arises from the use of convex hulls for heuristic collision-free motion planning of general objects among obstacles. Motion planning is easier for convex objects and obstacles, see for e.g., [19].

Several efficient algorithms for computing the convex hull of simple planar polygons are known, see for eg [39, 70]. These algorithms for planar polygons are iterative and vertex–based, i.e., the computation in each step depends on the region where the next vertex lies. By generalizing to an edge–based algorithm, [82] extended the planar polygon results to an algorithm for planar geometric models. Paper [93] also suggests an efficient convex hull algorithm based on a *bounding polygon* approach. In [21] another efficient algorithm is presented for computing the convex hulls of objects bounded by algebraic curves. The algorithm, which is partly a generalization of [59], reflects various practical considerations such as simplicity of implementation and flexibility to heuristic modifications.

A number of subtasks are again required to construct the convex hull of a curved model. All the subtasks required for the decomposition computation of planar geometric models, listed in the previous section, are also needed here. Besides these geometric operations, a monontone segmentation of the boundary curves of the planar model is also computed, as a pre-processing step to convex hull generation. The monotone segmentation requires adding singular points, flex points and extreme points on the boundary curves as extra vertices, [21]. Singularities, flex points and extreme point computations require straightforward computation of zero-dimensional point solutions of systems of polynomial equations, cf. Section 4.

Numerous algorithms for computing the convex hull of simple polyhedra have also been presented. See [39, 70] for a discussion of "beneath-and-beyond" and "divide-and-conquer" algorithms for computing the convex hull of point sets in three (and higher) dimensions. No significant theoretical advantage seems to be gained in knowing that these points are the vertices of a simple polyhedra [90]. Simplification in a practical setting, have yet to be addressed. As far as I am aware an algorithmic study of computing convex hulls of algebraic boundary models in three space is yet to be undertaken.

9 Sweeping and Convolution Operations on Models

There exists numerous applications such as automated assembly, numerical machining and part tolerancing, where generating the convolution or sweep of two curved models has proved useful. Motion planning in sophisticated modeling environments, for product prototyping and simulation also suggests the need to efficiently generate the surface boundary of sweeps of curved models [16]. For example, the motion of two objects in continuous contact in three-dimensional space can be represented as a point constrained to move on the boundary of convoluted solids [61, 84]. Generating various curves on the boundary of the convoluted solids then gives a way of generating motion paths, along which the given objects shall always keep in contact [19].

Efficient algorithms are known for generating the convolution of two convex polygons and polyhedra using methods for efficiently computing convex hulls, [61], and for the Minkowski sum, [48]. A convolution algorithm for two arbitrary planar geometric models, is described in [18] and for two convex, algebraic boundary models in [19]. See Figure 9.1, which shows the boundary (minus dangling edges) of the convolution of two planar curved models.

The problem of convoluting two algebraic boundary models, consists of a number of convolution subtasks with algebraic curves and surfaces as listed below. These in turn reduce to the computation of solutions of systems of polynomial equations, cf. Section 4. First there is the pairwise convolution of curve and surface elements of the two solids. In particular, the convolution of :

- two curved edges
- an edge and a face
- two faces.

The pairwise convolution needs to be computed for only those curve and surface elements, one from each solid, which have similar normal directions at some internal points. These compatible pairs of boundary elements can be computed by overlaying the tangent space Gaussian models of the two solids to be convoluted, see [19, 20]. The convolution surface generation is next followed by the computation of curve and surface singularities on the convoluted boundary. The analysis of these singularities is critical for the final topological reconstruction of the convolution boundary description.

The convolution of a sphere with an algebraic boundary model, alternatively the sweep of the sphere along all points on the boundary, is the same as constant radius "offsetting" of the model. Offsetting, one of the more important operations in geometric modeling because of immediate application in NC machining, has been considered by many authors recently. Paper [40] outlines exact offset procedures for convex polyhedra, convex solids of revolution and convex solids of linear extrusion. Paper [77] describe offsetting operations for solids represented in a dual form (boundary representation and constructive solid geometry), where objects are constructed from primitive solids which are natural quadrics. Paper [20] characterizes the offsetting problem for algebraic surfaces and provides an algebraic algorithm for its computations. This algorithm is based on such operations as computing resultants of polynomials, representing surface patches unambiguously, intersecting two algebraic surfaces and detecting self-intersections of algebraic surfaces. The efficiency of these operations however are quite limited for very high degree surfaces.

10 Conclusion

In the last few sections we presented brief descriptions of various geometric operations on physical object models with algebraic surfaces. However we also omitted discussing a large number of additional geometric operations, which are demanded by numerous applications using geometric modeling systems. We provide a shortlist of some of these here, with the hope that further attention will be paid to them in the future.

1. Area and volume computations.
2. Surface and volume mesh generation.
3. Graphics display and animation techniques.
4. Languages and geometric editing techniques.
5. Approximations, surface continuity and interpolation schemes.
6. Parallel algorithms.

Clearly much remains to be researched in the unfolding science of geometric modeling with algebraic surfaces.

Acknowledgements

The author owes much to the technical collaboration with his teachers, colleagues, friends and students. In particular, S. Abhyankar, J. Canny, T. Garrity, C. Hoffmann, J. Hopcroft, M. Kim, R. Lynch, J. Warren.

11 References

1. Abhyankar, S. S., (1971) Algebraic Space Curves, *Les Presses de L'Universite' de Montreal*, Montreal, Canada.
2. Abhyankar, S. S., (1976) Historical Ramblings in Algebraic Geometry and Related Algebra, *American Mathematical Monthly*, 83, 6, 409-448
3. Abhyankar, S. S., (1983) Desingularization of Plane Curves, *Proc. of the Symp. in Pure Mathematics*, 40, 1, 1-45.
4. Abhyankar, S., and Bajaj, C., (1987a), Automatic Rational Parameterization of Curves and Surfaces I: Conics and Conicoids, *Computer Aided Design*, 19, 1, 11-14.

5. Abhyankar, S., and Bajaj, C., (1987b), Automatic Rational Parameterization of Curves and Surfaces II: Cubics and Cubicoids, *Computer Aided Design*, 19, 9, 499-502.

6. Abhyankar, S., and Bajaj, C., (1988a), Automatic Rational Parameterization of Curves and Surfaces III: Algebraic Plane Curves, *Computer Aided Geometric Design*, 5, 309-321.

7. Abhyankar, S., and Bajaj, C., (1988b), Automatic Rational Parameterization of Curves and Surfaces IV: Algebraic Space Curves, Comp. Science Tech. Rept. 703 and CAPO report CER-87-5, Purdue University, *Presented at the 1987 SIAM Conference on Applied Geometry*, Albany, NY.

8. Abhyankar, S., and Bajaj, C., (1988c), Computations with Algebraic Curves, Comp. Science Tech. Rept. 806 and CAPO report CER-88-31, Purdue University, *Proceedings of the 1988 Intl. Symposium on Symbolic and Algebraic Computation, ISSAC-88*, Rome, Italy, to appear.

9. Arnon, D.S., and McCallum, S., (1983), A Polynomial-time Algorithm for the Topological Type of a Real Algebraic Curve, Tech. Rept. 454, Comp. Science Dept., Purdue University. See also *J. of Symb. Comput. 3:2*, 1988.

10. Bajaj, C., (1987), Algorithmic Implicitization of Algebraic Curves and Surfaces, Comp. Science Tech. Rept. 742 and CAPO report CER-88-13, Purdue University, *Presented at the 1987 Research Conference on Geometric Design*, Detroit, MI.

11. Bajaj, C., (1988a), Mathematical Techniques in Solid Modeling, *Proc. of the First International Conference on Computer Integrated Manufacturing*, RPI, Troy, 290-295.

12. Bajaj, C., (1988b), Approximation Methods for Algebraic Curves and Surfaces, CAPO report CER-88-36, Computer Science, Purdue University, *Presented at the 1988 Workshop on Algorithmic Aspects of Geometry and Algebra*, Mathematics Science Institute, Cornell University, Ithaca, New York.

13. Bajaj, C., (1988c), Towards Parameterizing Algebraic Varieties, *Manuscript*.

14. Bajaj, C., Canny, J., Garrity, T., and Warren, J., (1988), Decomposing Algebraic Sets, *Manuscript*.

15. Bajaj, C., Garrity, T., and Warren, J., (1988), On the Applications of Multi-equational Resultants, CAPO report CER-88-39, Computer Science, Purdue University,

16. Bajaj, C., Dyksen, W., Hoffmann, C., Houstis, E., and Rice, J., (1987) *Computing About Physical Objects*, Computer Science Technical Report, CAPO-87-1, Purdue University. *Proceedings of the 12th IMACS World Congress*, Paris, France, (1988), 642 - 644.

17. Bajaj, C., Hoffmann, C., Hopcroft, J. and Lynch, R., (1988), Tracing Surface Intersections, *Computer Aided Geometric Design*, 5, 285-307.

18. Bajaj, C., and Kim, M., (1987a) Generation of Configuration Space Obstacles : The Case of Moving Algebraic Curves, *Proc. of 1987 IEEE Conference on Robotics and Automation*, Raleigh, North Carolina, 979-984. Updated Version to appear in *Algorithmica*.

19. Bajaj, C., and Kim, M., (1987b) Compliant Motion Planning with Geometric Models, *Proc. of the Third ACM Symposium on Computational Geometry*, Waterloo, Canada, 171-180. Updated Version with title "Generation of Configuration Space Obstacles : The Case of Moving Algebraic Surfaces" to appear in *Intl. J. of Robotics Research*.

20. Bajaj, C., and Kim M., (1988a), Generation of Configuration Space Obstacles : The Case of Moving Spheres, *IEEE Journal on Robotics and Automation*, 4, 1, 94 - 99.

21. Bajaj, C., and Kim, M., (1988b), Algorithms for Planar Geometric Models, *Proceedings of the 15th International Colloquium on Automata, Languages and Programming (ICALP 88), Tampere, Finland, Lecture Notes in Computer Science*, Springer-Verlag 317, pp. 67–81

22. Bajaj, C., and Royappa, A., (1987), A Note on an Implementation of Sylvester's Resultant for Multivariate Polynomials, Comp. Science Tech. Rept. 718 and CAPO report CER-87-9, Purdue University.

23. Braid, I., Hillyard, R., and Stroud, I., (1980) Stepwise Construction of Polyhedra in Geometric Modeling, in *Mathematical Methods in Computer Graphics and Design*, eds., K. Brodie, Academic Press, New York.

24. Blythe, W., (1905) *On Models of Cubic Surfaces*, Cambridge University Press.

25. Boehm, W., Farin, G., and Kahmann, J., (1984), A Survey of Curve and Surface Methods in CAGD, *Computer Aided Geometric Design*, 1, 1-60.

26. Brent, R., Gustavson, F., and Yun, D., (1980), Fast Solutions of Toeplitz Systems of Polynomial Equations and Computation of Pade' Approximants, *Journal of Algorithms*, 1, 259 - 295.

27. Buchberger, B., (1984) Gröbner Bases: An Algorithmic Method in Polynomial Ideal Theory, in *Recent Trends in Multidimensional System Theory*, N. Bose (eds)., Reidel.

28. Canny, J., (1988), The Complexity of Robot Motion Planning, ACM Doctoral Dissertation Series, MIT Press, Cambridge, Mass.

29. Canny, J., (1988), Generalized Characteristic Polynomials, *Proceedings of the 1988 Intl. Symposium on Symbolic and Algebraic Computation, ISSAC-88*, Rome, Italy.

30. Chazelle, B., (1984), Convex Partitions of Polyhedra: A lower bound and worst-case optimal algorithm, *SIAM J. on Computing*, Vol. 13, No. 3, pp. 488–507.

31. Clarkson, K., Edelsbrunner, H., Guibas, L., Sharir, M., Welzl, E., (1988), Combinatorial Complexity Bounds for Arrangements of Curves and Surfacces, *Proc. of the 29th Foundations of Computer Science*, 568 - 579.

32. Cohen, E., Lyche, T., and Riesenfeld, R., (1980), Discrete B-splines and Subdivision Techniques in Computer Aided Geometric Design and Computer Graphics, *Computer Graphics and Image Processing*, 14, 87 - 111.

33. Collins, G., (1971), The Calculation of Multivariate Polynomial Resultants, *Journal of the ACM*, 18, 4, 515 - 532.

34. Collins, G., (1983), Quantifier Elimination for Real Closed Fields: A Guide to the Literature, in *Computer Algebra, Symbolic and Algebraic Computation*, 2nd Edition, B. Buchberger, G. Collins, R. Loos, eds., Springer Verlag, 1983, 79 - 82.

35. de Boor, C., (1978), *A Practical Guide to Splines*, AMS 27, Springer Verlag.

36. de Montaudouin, Y., and Tiller, W., (1984), The Cayley Method in Computer Aided Geometric Design, *Computer Aided Geometric Design*, 1, 4, 309 - 326.

37. de Montaudouin, Y., and Tiller, W., (1986), Applications of Power Series in Computational Geometry, *Computer Aided Design*, 18, 10, 514 - 524.

38. Dobkin, D.P., Souvaine, D.L., and Van Wyk, C.J., (1986), Decomposition and Intersection of Simple Splinegons, *Algorithmica*, 3, 473 - 485.

39. Edelsbrunner, H., (1987) *Algorithms in Combinatorial Geometry*, Springer Verlag.

40. Farouki, R., (1985), Exact Offset Procedures for Simple Solids, *Computer Aided Geometric Design*, 2, 4, 257 - 259.

41. Farouki, R., (1988), Computational Issues in Solid Boundary Evaluation, *Information Sciences*, 44, 87 - 171.

42. Farouki, R., and Rajan, V., (1988), On the Numerical Condition of Polynomials in Bernstein Form, *Computer Aided Geometric Design*, to appear.

43. Garrity, T., and Warren, J., (1988), On Computing the Intersection of a Pair of Algebraic Surfaces, *Computer Aided Geometric Design*, to appear.

44. Geisow, A., (1983), *Surface Interrogations*, Ph.D. Thesis, University of East Anglia, School of Computing Studies and Accountancy.

45. Goldman, R., Sederberg, T., and Anderson, D., (1984), Vector Elimination: A Technique for the Implicitization, Inversion and Intersection of Planar Parametric Rational Polynomial Curves, *Computer Aided Geometric Design*, 1, 327 - 356.

46. Gordon, W., and Riesenfeld, R., (1974) Bernstein-Bezier Methods for Computer Aided Geometric Design of Free Form Curves and Surfaces, *Journal Of ACM*, 21, 2, 293 -310.

47. Gordon, W., and Riesenfeld, R., (1974) B-spline Curves and Surfaces, *Computer Aided Geometric Design*, eds., R. Barnhill and R. Riesenfeld, Academic Press.

48. Guibas, L., and Seidel, R., (1987) Computing Convolutions by Reciprocal Search, *Discrete and Computational Geometry*, Vol. 2, No. 2, pp. 175–193.

49. Hilbert, D, and Cohn-Vossen, S., (1932) *Geometry and the Imagination*, Chelsea Publishing Co., New York.

50. Hoffmann, C., and Hopcroft, J., (1986), Quadratic Blending Surfaces, *Computer Aided Design*, 18, 6, 301 - 306.

51. Hoffmann, C., and Hopcroft, J., (1987), Geometric Ambiguities in Boundary Representations, *Computer Aided Design*, 19, 3, 141 - 147.

52. Hoffmann, C., Hopcroft, J., and Karasick, M., (1988), Robust Operations for Polyhedral Models, *Proc. of the Fourth ACM Symposium on Computational Geometry*, Urbana, Illinois, 106-118.

53. Henderson, A., (1911) *The Twenty Seven Lines upon the Cubic Surface*, Cambridge Tracts in Math. And Math. Physics, no 13.

54. Jenkins, M., and Traub, J., (1970) A Three-stage algorithm for Real Polynomials using Quadratic Iteration, *Communications of the ACM*, 15, 97 - 99.

55. Jenkins, M., and Traub, J., (1972) Zeros of a Complex Polynomial, *Siam J. on Numerical Analysis*, 7, 545 - 566.

56. Johnstone, J., and Bajaj, C., (1988), On the Sorting of Points along an Algebraic Curve, CAPO report CER-88-37, Computer Science, Purdue University,

57. Karasick, M., (1988), *On the Representation and Manipulation of Rigid Solids*, Ph.D. Thesis, McGill University, Department of Computer Science.

58. Lazard, D., (1981), Resolution des Systemes d'Equations Algebriques, *Theoretical Computer Science*, 15, 77 - 110.

59. Lee, D.T., (1983), On Finding the Convex Hull of a Simple Polygon, *International Journal of Computer and Information Sciences*, Vol. 12, No. 2, pp. 87–98.

60. Levin, J., (1979), Mathematical Models for Determining the Intersections of Quadric Surfaces, *Computer Graphics and Image Processing*, 11, 73 - 87.

61. Lozano-Pérez, T., (1983) Spatial Planning: A Configuration Space Approach, *IEEE Trans. on Computers*, Vol.C-32, pp. 108–120.

62. Macaulay, F., (1902), Some Formulae in Elimination, *Proc. London Mathematical Society*, 1, 35, 3 - 27.

63. Martin, R., de Pont, J., and Sharrock, T., (1986) Cyclide Surfaces in Computer Aided Design, *The Mathematics of Surfaces*, ed. J. Gregory, Oxford University Press, 253 - 268.

64. Newmann, W., and Sproull, R., (1979), *Principles of Interactive Computer Graphics*, McGraw Hill, New York.

65. Ocken, S., Schwartz, J., and Sharir, M., (1983), Precise Implementation of CAD Primitives using Rational Parameterizations of Standard Surfaces, *Planning, Geometry and Complexity of Robot Motion*, Ablex Publishing, Norwood, N.J., 245–266.

66. Patterson, R., (1988) Parametric Cubics as Algebraic Curves, *Computer Aided Geometric Design*, 5, 139 - 159.

67. Pogorelov, A.V., (1978), *The Minkowski Multidimensional Problem*, Winston, New York

68. Pratt, M., (1986), Parametric Curves and Surfaces as used in Computer Aided Design, *The Mathematics of Surfaces*, ed. J. Gregory, Oxford University Press, 117–142.

69. Pratt, M., and Geisow, A., (1986), Surface/Surface Intersection Problems, *The Mathematics of Surfaces*, ed. J. Gregory, Oxford University Press, 117 - 142.

70. Preparata, F., and Shamos, M., (1985) *Computational Geometry, An Introduction* Springer Verlag.

71. Requicha, A., (1980), Representations of Rigid Solid Objects, *Springer Lecture Notes in Computer Science*, Vol. 89, pp. 2–78.

72. Requicha, A., and Voelcker, H., (1983), Solid Modeling: Current Status and Research Directions, *IEEE Comp. Graphics and Applic.*, 3, 25 - 37.

73. Requicha, A., and Voelcker, H., (1985), Boolean Operations in Solid Modeling: Boundary Evaluation and Merging Algorithms, *Proc. of IEEE*, 73, 1, 30 - 44.

74. Renegar, J., (1987a), On the Worst Case Arithmetic Complexity of Approximating Zeros of Polynomials, *Journal of Complexity*, 3, 90 - 113.

75. Renegar, J., (1987b), On the Worst Case Arithmetic Complexity of Approximating Zeros of Systems of Polynomials, Technical Report, Operations Research and Industrial Engineering, Cornell University.

76. Rowe, J., (1917), The Equation of a Rational Plane Curve Derived from its Parametric Equations, *Bulletin of American Mathematical Society*, 304 - 308.

77. Rossignac, J., and Requicha, A., (1986), Offsetting Operations in Solid Modeling, *Computer Aided Geometric Design*, 3, 2, 129 - 148.

78. Saia, A., Bloor, M., and De Pennington, A., (1987) Sculptured Solids in a CSG based Geometric Modeling System, *The Mathematics of Surfaces II*, ed. R. Martin, Oxford University Press, 321 - 342.

79. Sabin, M., (1986), Recursive Division. *The Mathematics of Surfaces*, ed. J. Gregory, Oxford University Press, 269 - 282.

80. Salmon, G., (1885), *Lessons Introductory to the Modern Higher Algebra*, Reprinted, Chelsea Publishing.

81. Salmon, G., (1914), *A Treatise on the Analytic Geometry of Three Dimensions*, vol I, II, ed., R. Rogers. Reprinted, Chelsea Publishing.

82. Schäffer, A., and Van Wyk C., (1987), Convex Hulls of Piecewise-Smooth Jordan Curves, *Journal of Algorithms*, Vol. 8, No. 1, pp. 66–94.

83. Schwartz, J., (1980), Fast Probabilistic Algorithms for Verification of Polynomial Identities, *Journal of the ACM*, 27, 4, 701 - 717.

84. Schwartz, J. and Sharir, M., (1983), On the Piano Movers Problem: II, General Techniques for Computing Topological Properties of Real Algebraic Manifolds, *Adv. Appl. Math.* Vol. 4, pp. 298–351.

85. Sederberg, T., (1985) Piecewise Algebraic Surface Patches, *Computer Aided Geometric Design*, 2, 53 - 59

86. Sederberg, T., (1986) Improper Parameterization of Rational Curves, *Computer Aided Geometric Design*, 3, 67 - 75

87. Sederberg, T., Anderson, D., and Goldman, R., (1984) Implicit Representation of Parametric Curves and Surfaces, *Computer Vision, Graphics and Image processing*, 28, 72 - 84.

88. Sederberg, T., and Snively, J., (1987), Parameterization of Cubic Algebraic Surfaces, *The Mathematics of Surfaces II*, ed. R. Martin, Oxford University Press, 299 - 320.

89. Segre, B., (1942) *The Non-singular Cubic Surfaces*, Oxford at the Clarendon Press.

90. Seidel, R., (1985) A Method for Proving Lower Bounds for certain Geometric Problems, in *Computational Geometry*, ed. G. Toussaint, North Holland, 319 - 334.

91. Snyder, V., and Sisam, C., (1914), *Analytic Geometry of Space*, Henry Holt and Company, New York.

92. Snyder, V., Coble, A., Emch, A., Lefschetz, Sharpe, F., and Sisam, C., (1928), *Selected Topics in Algebraic Geometry*, Reprinted, Chelsea Publishing 1970.

93. Souvaine. D.L., (1986), *Computational Geometry in a Curved World*, Ph.D Thesis, Computer Science Technical Report CS-TR-094-87, Princeton University.

94. Sugihara, K., (1987) On the Finite Precision Representations of Geometric Objects, Research Memorandum RMI 87-06, Department of Mathematical Engineering and Instrumentation Physics. Tokyo University.

95. Sugihara, K., and Iri, M., (1988) Geometric Algorithms in Finite Precision Arithmetic, Research Memorandum RMI 88-10, Department of Mathematical Engineering and Instrumentation Physics, Tokyo University.

96. Tarjan, R.E., and Van Wyk, C.J., (1988), An $O(n \log \log n)$-Time Algorithm for Triangulating Simple Polygons, *SIAM Journal on Computing*, Vol. 17, No. 1, pp. 143–178.

97. van der Waerden, B., (1938), *Einführung in die algebraische Geometrie*, Springer Verlag, 2nd edition, 1973

98. van der Waerden, B., (1950), *Modern Algebra*, vol. II, Ungar Publishing, New York.

99. Walker, R., (1978), *Algebraic Curves*, Springer Verlag, New York.

100. Warren, J., (1987), Blending Quadric Surfaces with Quadric and Cubic Surfaces, *Proc. of the Third ACM Symposium on Computational Geometry*, Waterloo, Canada, 341-353.

101. Weiler, K., (1985) Topological Structures for Geometric Modeling, Ph.D. Thesis, Rensselear Polytechnic Institute, Department of Computer and System Engineering.

102. Wesley, M., Lozano-Perez, T., Liberman, L., Lavin, M., and Grossman, D., (1980), A Geometric Modeling System for Automated Mechanical Assembly, *IBM J. Res. and Dev.*, 24, 64 - 74.

103. Wu, M., Bajaj, C., Liu, R., (1988), Face Area Evaluation Algorithm for Solids, *Computer Aided Design*, 20, 2, 75-82.

104. Zariski, O., (1935), *Algebraic Surfaces*, Ergebnisse der Mathematik und ihre Grenzgebiete, Vol. 4.

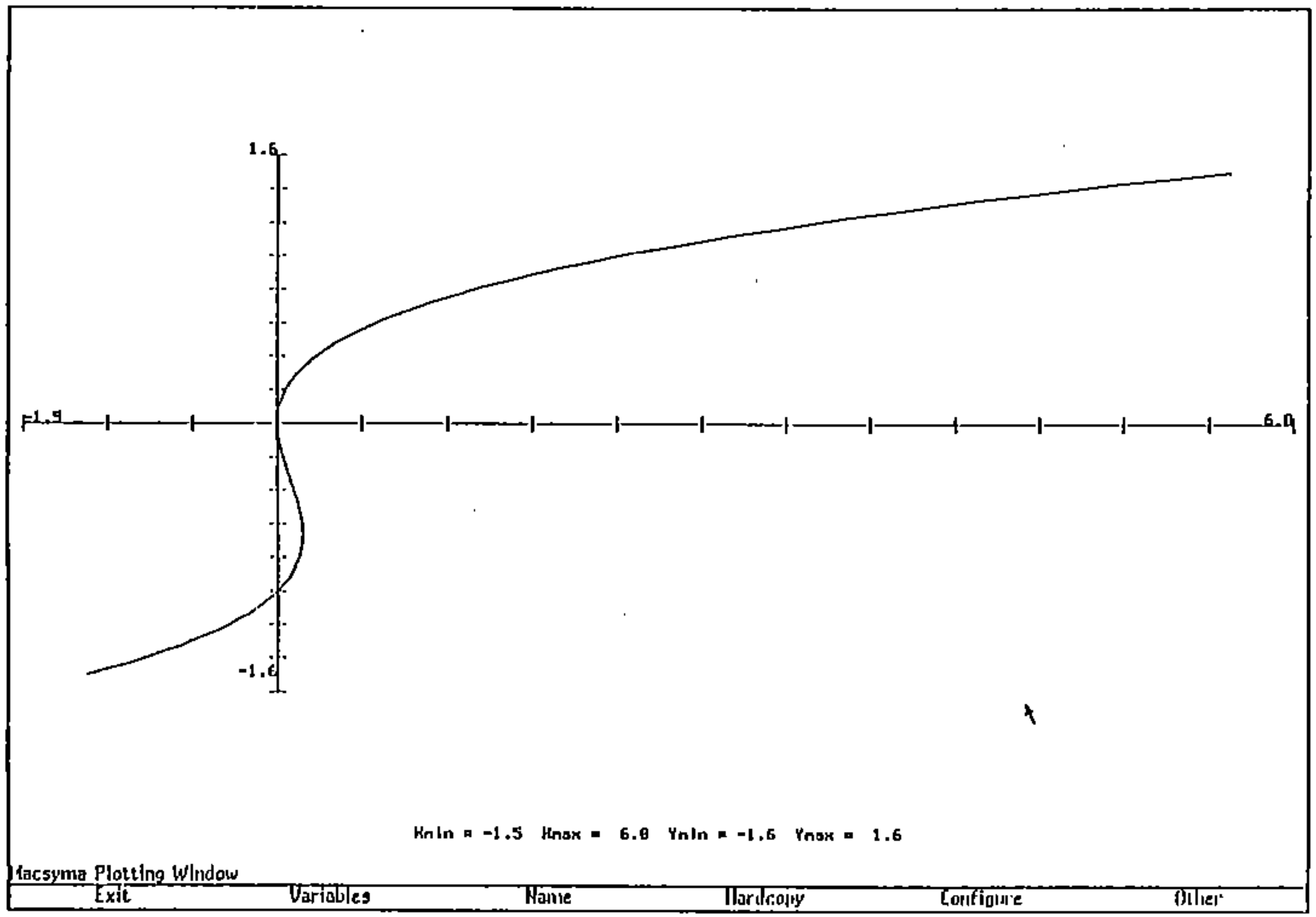

FIGURE 2.1.1: Rational Curve I

$$x - y^2 - y^3 = 0 \qquad x = t^2 + t^3$$
$$y = t$$

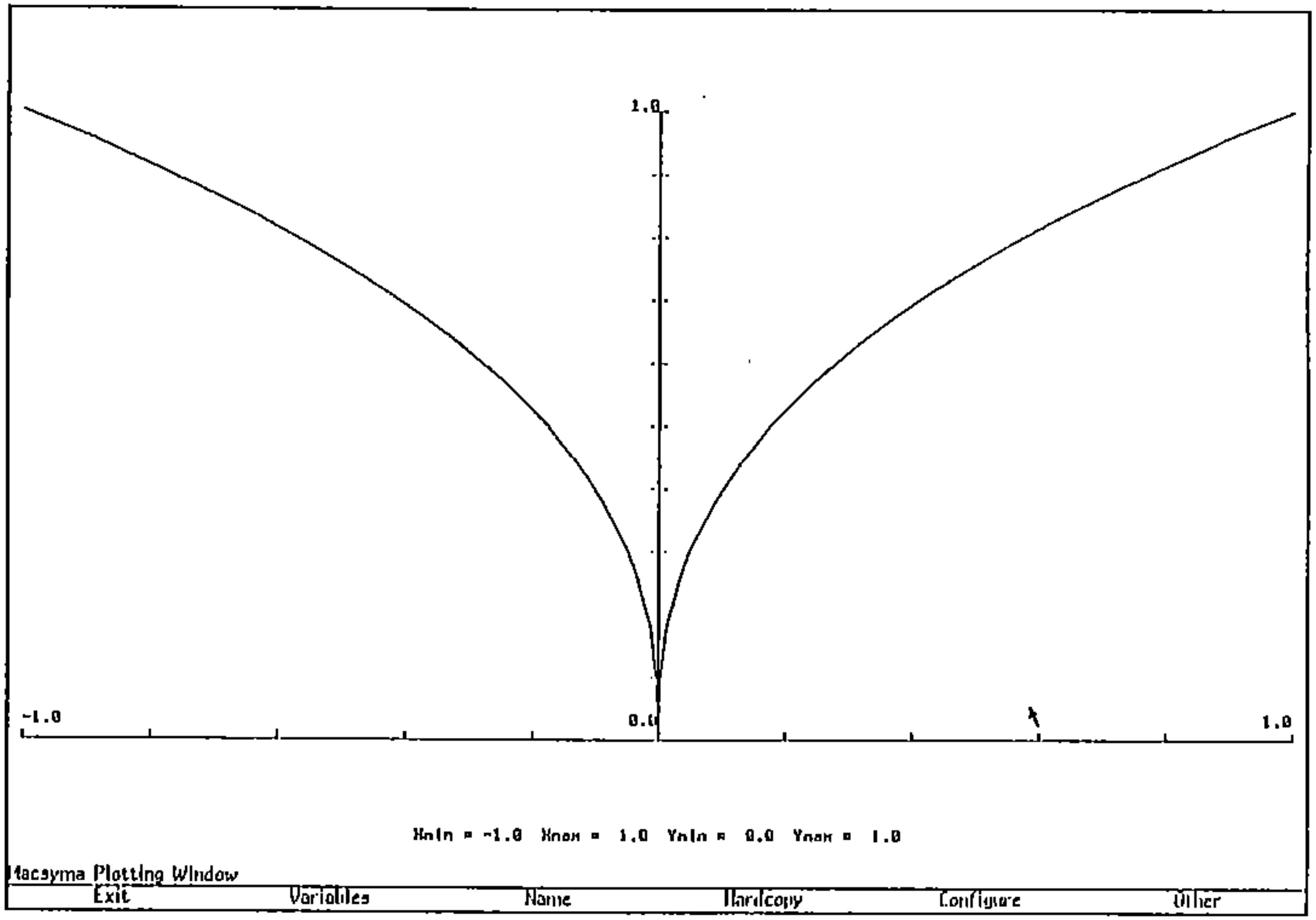

FIGURE 2.1.2: Rational Curve II

$x^2 - y^5 = 0$ $\qquad x = t^5$

$y = t^2$

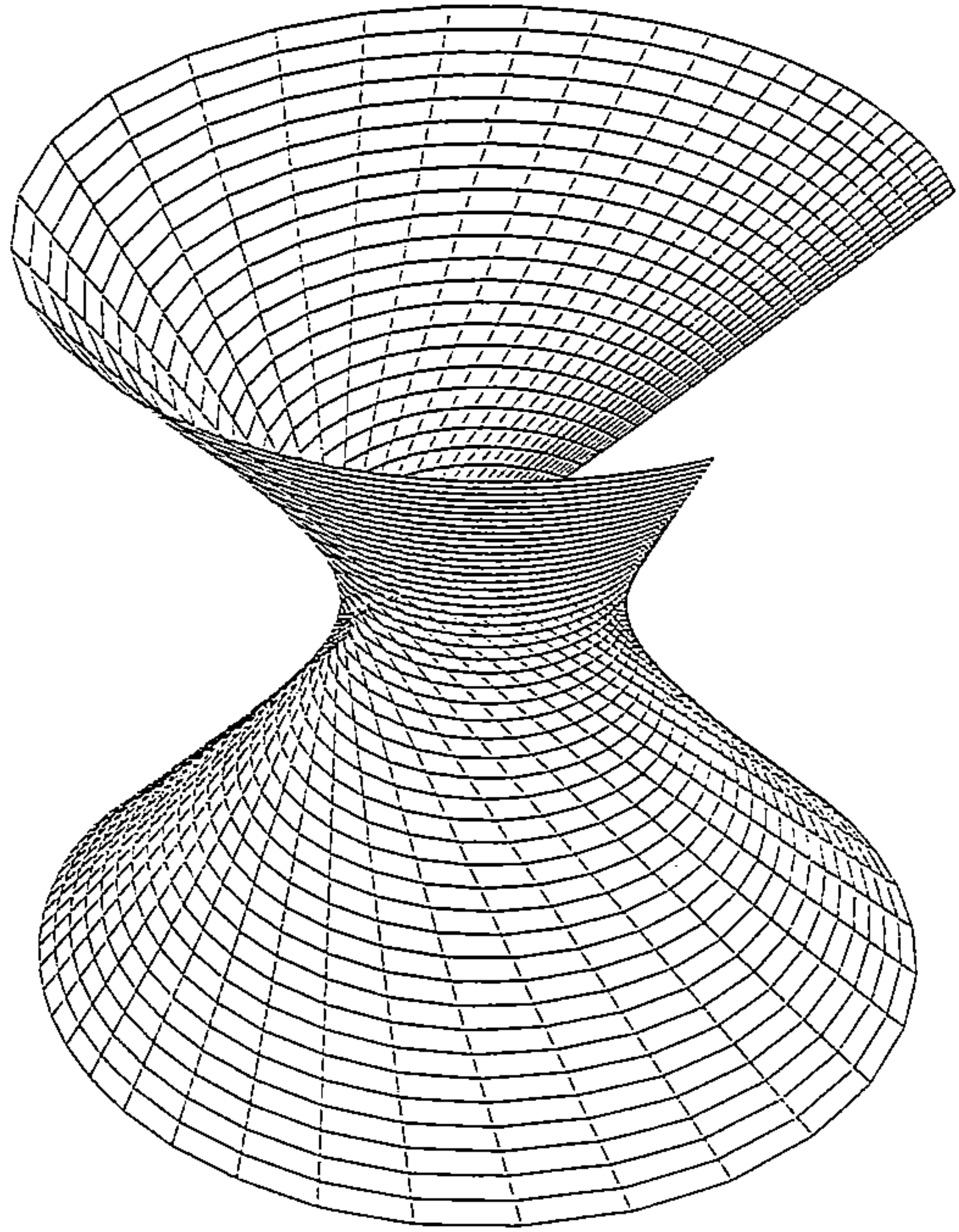

FIGURE 2.1.3: Rational Surface I

$x^2 + y^2 - z^2 - 1 = 0$ $\quad$ $x = t^2 - 2st - 1/t^2 + 1$

$y = [2t + s(t^2 - 1)] / t^2 + 1$

$z = s$

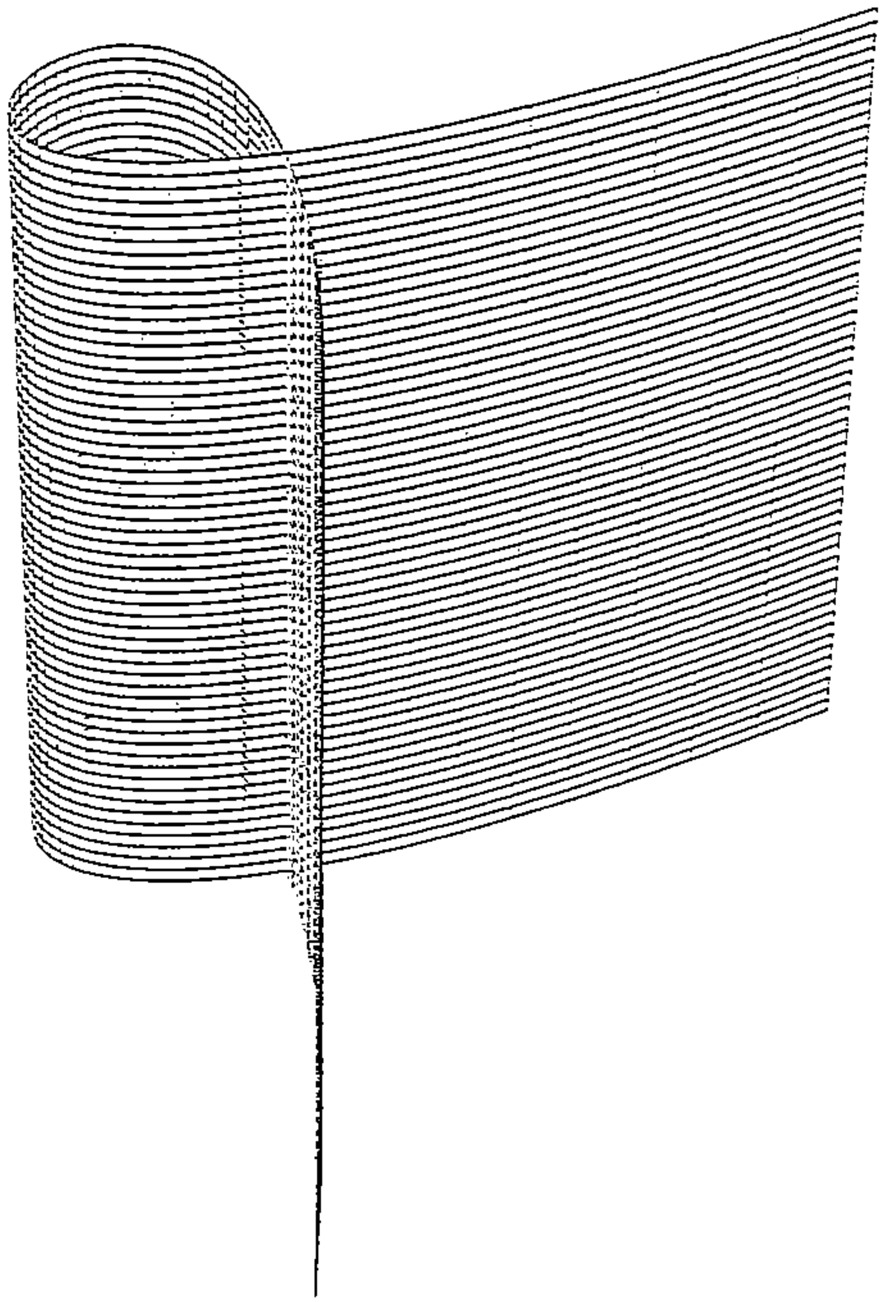

FIGURE 2.1.4: Rational Surface II

$x^2 - y^2 - y^3 = 0$ $\quad$ $x = t^3 - t$

$y = t^2 - 1$

$z = s$

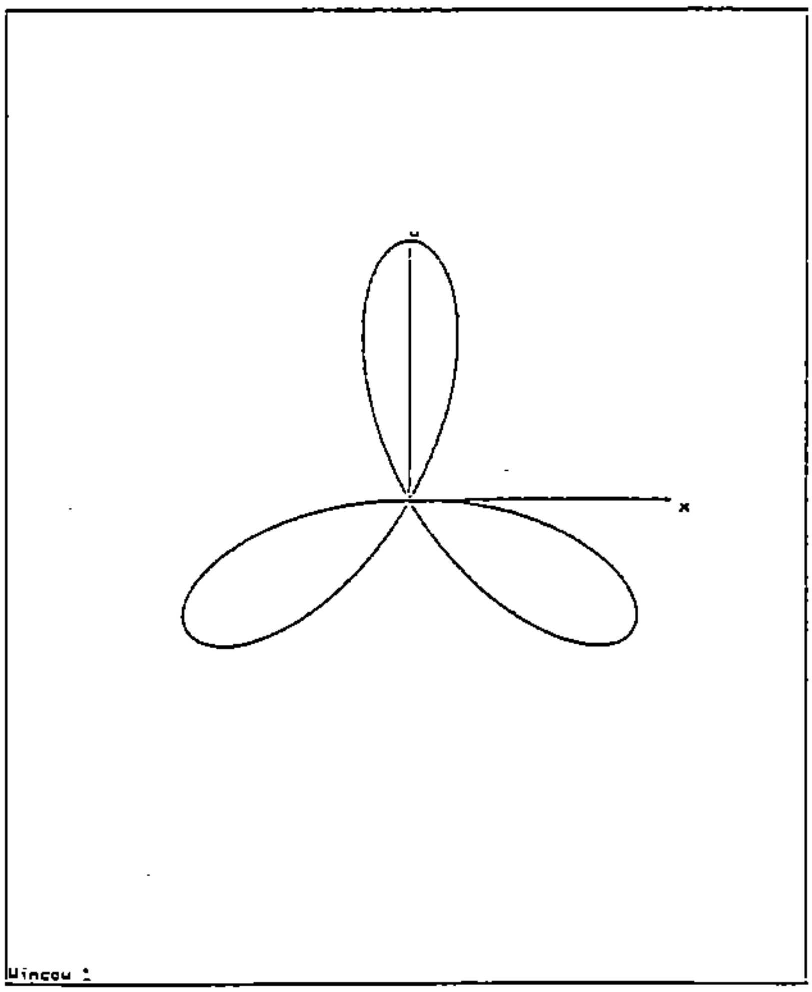

FIGURE 2.2.1: Singular Algebraic Plane Curve C_1
$x^4 + 2x^2y^2 + y^4 + 3x^2y - y^3 = 0$

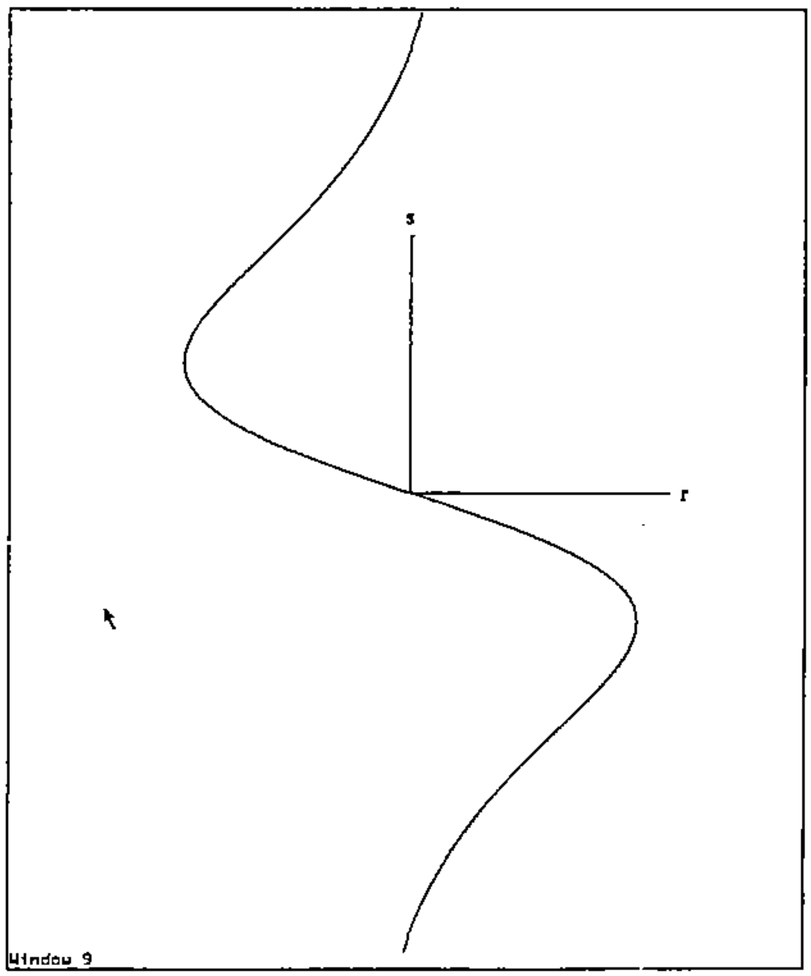

FIGURE 2.2.2: C_1 after a Quadratic Transformation
$rs^4 + 2rs^2 - s^3 + r + 3s = 0$

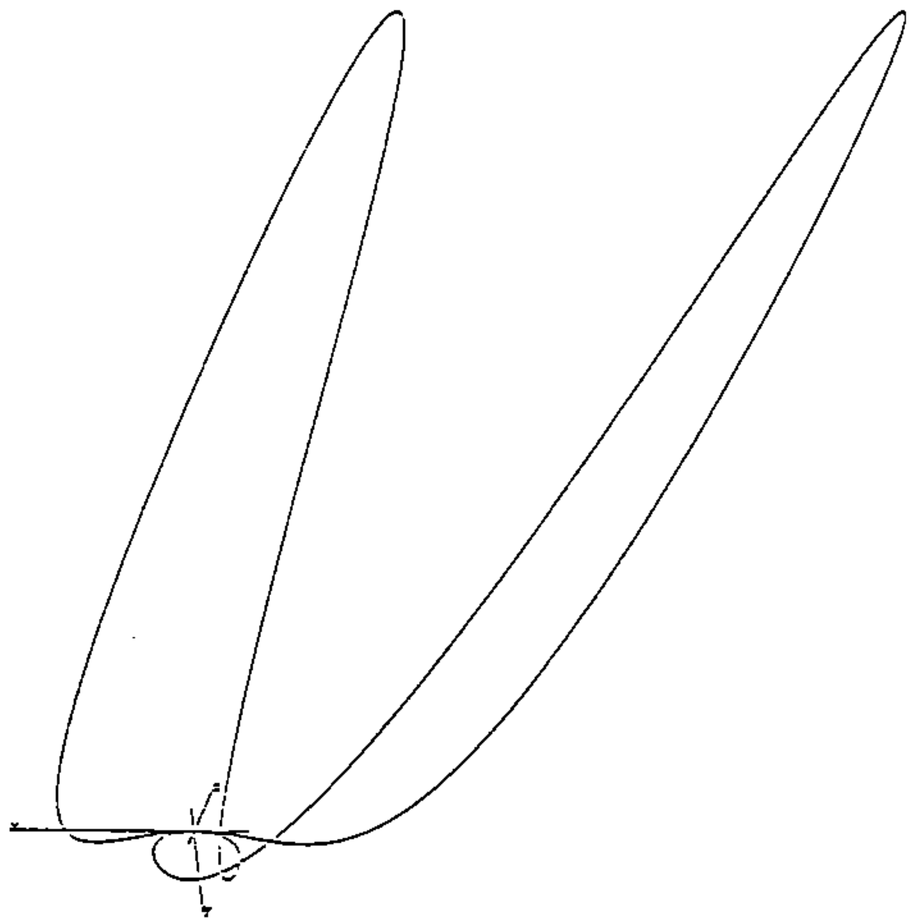

FIGURE 4.3.1: Numerically Traced Space Curve

$x = s$	$x = s$
$y = t$	$y = t$
$z = 2s^4 + t^4$	$z = 3s^2t - t^2 + 2t^3$

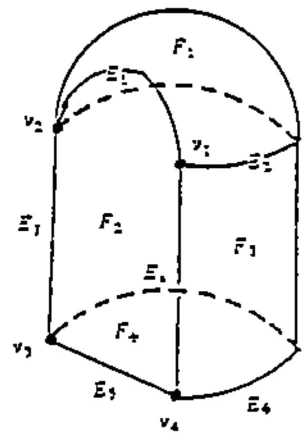

FIGURE 5.2.1: Algebraic Boundary Model

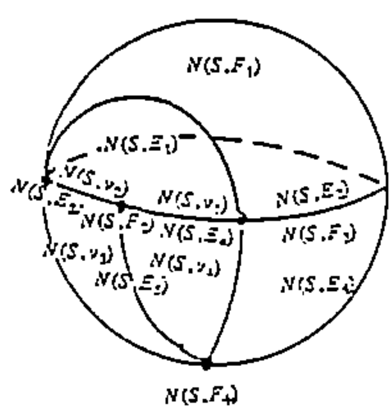

FIGURE 5.2.2: Gaussian Model

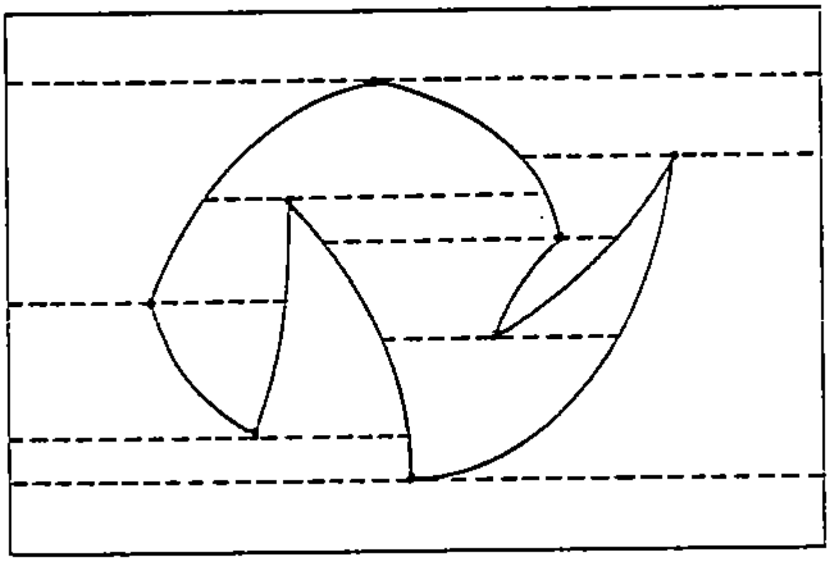

FIGURE 7.1: Horizontal Visibility Partition

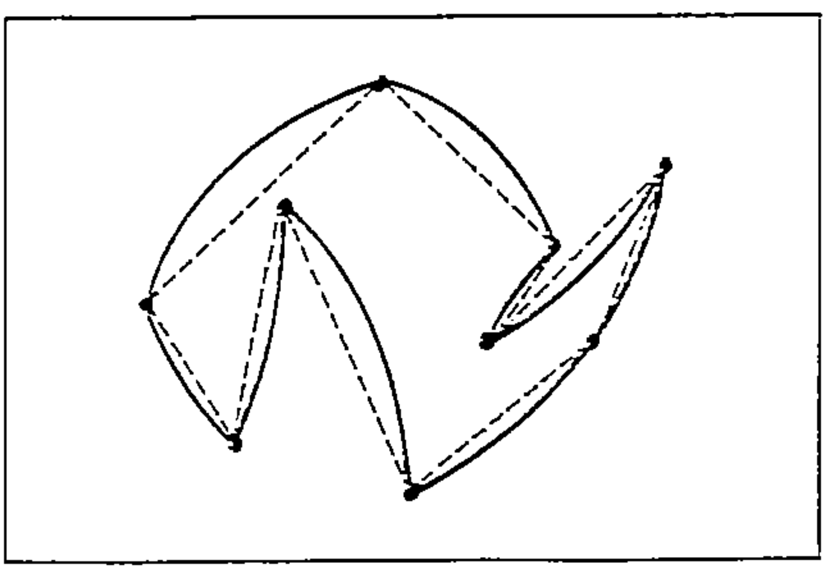

FIGURE 7.2: Characteristic Carrier Polygon

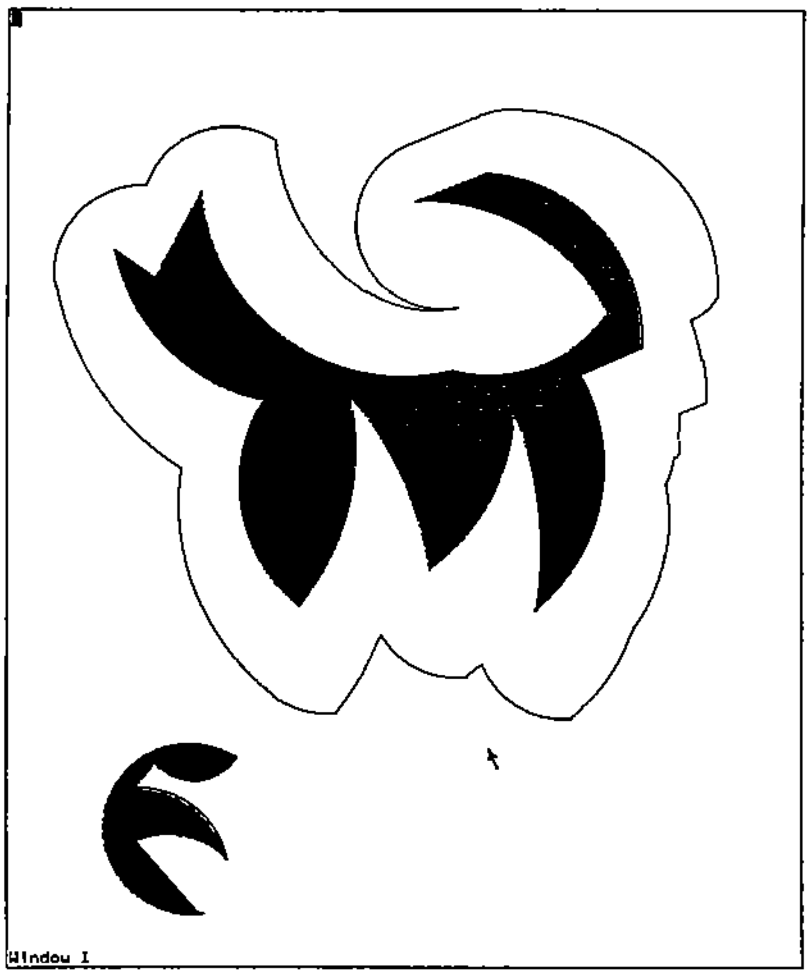

FIGURE 9.1: Convolution Boundary